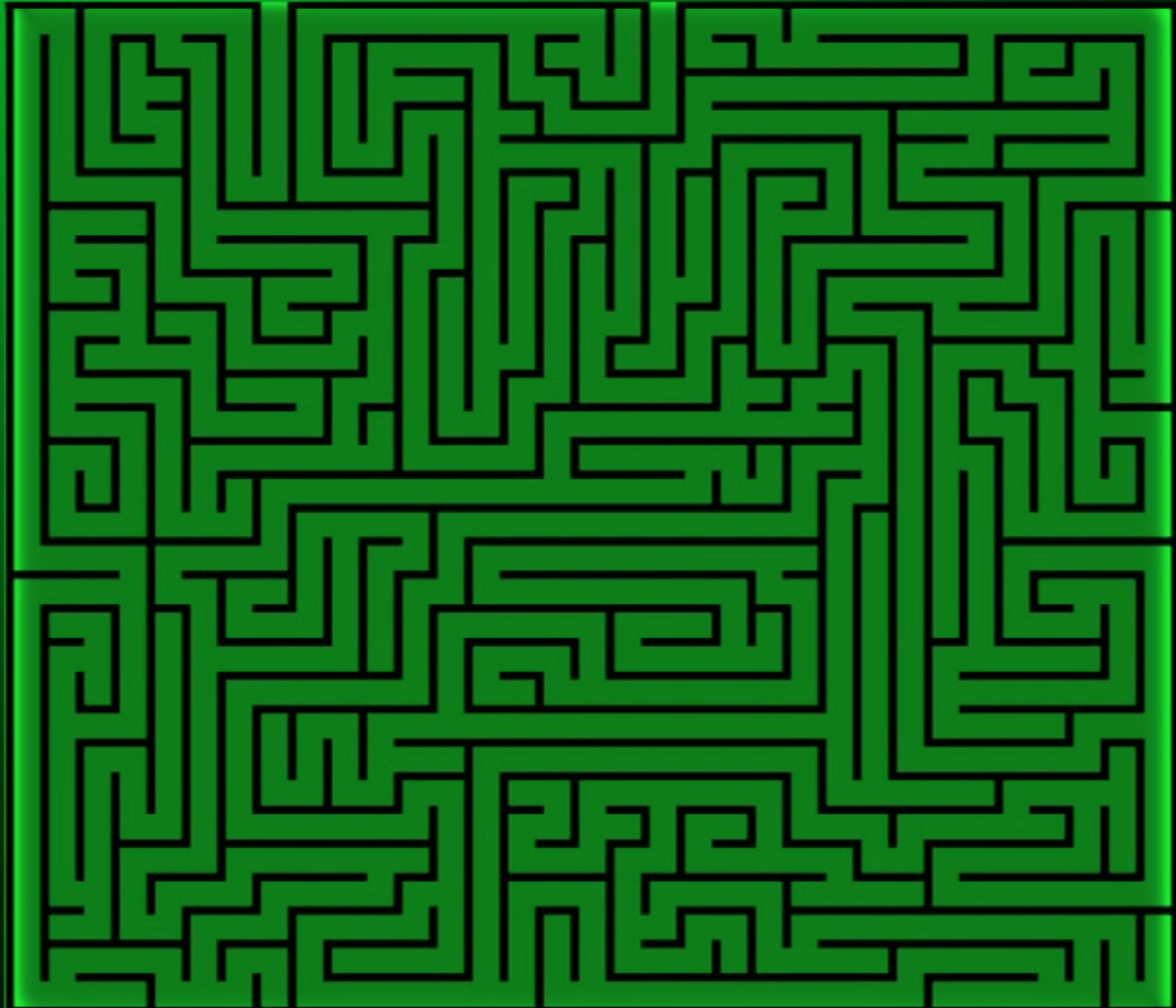


Club **HACK** Mag

1st Indian "**HACKING**" Magazine



It's amazing...What you can find, or lose on the network

Issue 10 | Nov 2010
www.clubhack.com

TechGyan NETWORK ATTACKS | **LegalGyan** ELECTRONIC CONTRACTS |
ToolGyan GREASE MONKEY | **Mom's Guide** WHOIS? QUERY |

Hi Friends, I am Aarja Bhattacharyya the cheerful content reviewer of this super cool hacking magazine. Hope my name rings a bell especially for two reasons. First the name itself and second since I am the only girl in this hacker's den. In case you have other reasons you may share it with us :) So getting the focus on to the much awaited magazine CHMag, you all must be wondering about the delay in the release of this issue. Actually it was deliberate since we wanted to release this issue on the auspicious occasion of the birthday of our founder i.e. the dad hacker (now officially true!!) Rohit Srivastwa.



Aarja Bhattacharyya

Here we are with our 10th issue of CHMag. Its amazing what you can find or loose on the network, the Tech Gyan will focus on network attacks, the Tool Gyan is on Greasemonkey- a customized way to browse internet, Moms Guide on whois query, Legal Gyan on electronic contracts and Command Line titled Let's zip it up. We all are geared up for ClubHack 2010 and very much excited for it, hope you have registered for it, if not what are you waiting for? christmas? :P here's where you can register: <http://clubhack.com/2010/registration/> hurry up! do not miss the chance to meet the International Security Guru- Bruce Schneier.

ClubHACKMag

Issue 10, November 2010.

Team CHmag

Rohit Srivastwa

rohit@clubhack.com

Aarja Bhattacharyya

aarja@chmag.in

Abhijeet R Patil

abhijeet@chmag.in

Abhishek Nagar

abhishek@chmag.in

Deepranjan S More

deepranjan@chmag.in

Pankit Thakkar

pankit@chmag.in

Varun V Hirve

varun@chmag.in

www.chmag.in

info@chmag.in

CONTENTS

Pg 03	TechGyan Network Attacks
Pg 07	ToolGyan Greasemonkey
Pg 14	Mom'sGuide Whois Query
Pg 16	LegalGyan Electronic Contracts
Pg 19	Command LineGyan Let's zip it up



Network Attacks

Introduction

Any method, technique or process used to attack and compromise the security of the network can be termed as a Network attack. There can be a number of motives behind the attacks like fame, terrorism, greed, etc. A few types of various malicious attacks are covered in this article.

Types of Network Attacks

The common and popular attacks would be:-

1. Eavesdropping
2. Denial-of-Service
3. Session Hijacking
4. IP Spoofing

5. DNS Spoofing
6. Man-in-the-Middle Attack

Eavesdropping

Eavesdropping is basically the act of secretly listening to the conversation of others, obviously without their permission. This definition can also be applied to network sniffing. In network sniffing, attacker secretly sniffs/listens to the data transmitted through the network.

The modules operating would be like this - A machine is configured to “listen” mode and then it is used to capture the juicy data from the network! This can be done using readily available programs like Cain and Abel, Ehtercap, SSLsniff, etc.

Denial-of- Service

Wikipedia: - A denial-of-service attack (DoS attack) is an attempt to make a computer resource unavailable to its intended respondents. It generally consists of the concerted efforts of a person or people to prevent an Internet site or service from functioning efficiently or at all, temporarily or indefinitely.

Methods of attacks

Smurf Attack

These attacks can be destructive. In this attack, an attacker sends a large amount of ICMP echo (ping) traffic at IP broadcast addresses. These packets have spoofed IP address of the source pointing to the victim. To amplify the attack several intermediary sites are selected by the attacker. This results in lots of ping replies (ICMP echo Reply) and thus resulting in victim being compromised.

SYN Flood attack

SYN flood attacks exploits TCP three-way handshake. In this attack, attacker sends lots of TCP SYN packets to the victim with spoofed source IP address. These packets try to establish connection with the victim. Now what happens is that the victim sends back a TCP SYN-ACK packet and waiting for the response from the source. But as the source address is spoofed the response never comes thus creating half open connections.

This floods the available connection with the server and in the process keeps the server from responding to legitimate traffic.

This flooding if done in large volume can cause DoS.

Distributed Denial-of-service (DDoS) Attack

In DDoS attack, the attacker compromises large number of computers, mostly in different locations. These compromised machines are called as secondary victims or Zombies. Then these zombies are used as attack platform to attack the primary victim.

The zombies or the secondary victims may not be aware that they are being used to attack the primary victim. Trojans and viruses give the control attacker to these machines to launch attacks of the victim.

This attack is difficult to detect as the attack comes from several IP address. This is the most deadly attack of all and not easy to overcome.

Session Hijacking

Session hijacking exploits computer session between two machines. Here, computer session means connection between two machines.

When a TCP session is established a cookie is used to verify if the session is active or not. The attacker can steal these cookies by sniffing or using the saved cookies on victim's computer. Since most of authentication is done only at the start of the session, this allows the hacker to assume the identity of the victim and gains the same access to the resources as that of the victim.

Types of Session Hijacking attacks: -

1. Active
2. Passive

In an Active attack, attacker hijacks an existing session on the network by doing a Man-in-the-middle attack. This allows the attacker to execute a various commands in

order to maintain his access, delete the traces etc. The attacker can create accounts on the network which can be used to gain access later without having to do session hijack every time.

In Passive attack, attacker monitors the ongoing session in the network. This attack uses sniffer tools to sniff around the network and find juicy information!

The third type, Hybrid attack, uses the combination of the above mentioned attacks. This attack is used to sniff and modify the data simultaneously.

IP Spoofing

IP spoofing, also known as IP address forgery, is a technique that replaces the original IP address with another machine's address in which an attacker impersonates as a trusted host to conceal his identity, spoof a Web site, hijack browsers, or gain access to a network.

Here's how it works: The attacker obtains the IP address of a legitimate host and alters IP packet headers so that the legitimate host appears to be the source. So now when a visitor types in a URL of a legitimate site, he is taken to a fraudulent web page created by the attacker. For example, if the attacker has spoofed a site, say www.abc.com, then any visitor who types this in the URL would see spoofed content created by the attacker instead of the original content.

With this kind of attack, the attacker could gain access to juicy information such as passwords, credit cards numbers, etc or install malware or alter the data.

DNS Spoofing

Domain Name Service (DNS) basically transforms a domain name, (say www.example.com) to its IP address (say 11.22.33.44). AND DNS spoofing is a technique where in a DNS entry to point to another IP rather than it is supposed to point to.

There are two methods of DNS spoofing:-

1. DNS Cache Spoofing: - DNS server cannot store information about all existing domain names and IP addresses in its cache. It is done to avoid constant repetitions of inquiries to login to servers of corresponding domains.

Now data is introduced into a DNS name server's cache database that did not originate from authoritative DNS sources. Its maliciously crafted attack on the name server. It may also result from improper software design of DNS applications.

The second variant of the attack directed on substitution DNS, consists in change of a server cache DNS.

2. DNS ID spoofing:- The heading of a package of the DNS-protocol contains an identification field for conformity of inquiries and answers. The purpose of substitution DNS ID is to send the answer to DNS-inquiry before the present DNS-server will answer. For performance of it, it is necessary to predict the identifier of inquiry. Locally it is realized by simple listening of the network traffic.

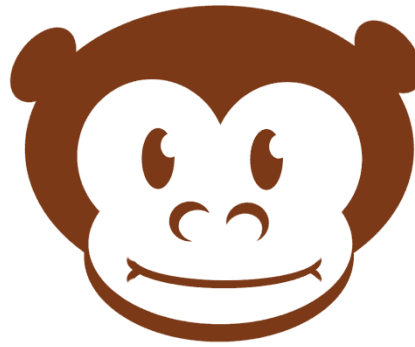
Man-in-the-Middle Attack

In Man-in-the-Middle (MITM) attack, the attacker intercepts the traffic between two machines and make the victims believe that they are talking directly to each other, when in fact their conversation is controlled by the attacker.

The attacks starts with sniffing and eavesdropping and after the attacker gains access to the conversation, he can extract juicy information like passwords, credit cards numbers, etc. or can alter the data, install malwares.



Abhijeet Patil
abhijeet@chmag.in



Greasemonkey - A customized way to browse internet

Introduction

I am writing this article by considering that my dear readers are well aware of Mozilla Firefox which may be the best web browser after Internet Explorer. As it always happens, a new thing emerges by introducing new concepts for users. Similarly Mozilla Firefox introduced new add-on to the internet world - Greasemonkey. You might be thinking what this monkey does for us? Answer is "Greasemonkey allows you to customize the way a webpage is displayed using small bits of JavaScript". Yes! This is a monkey who does this magic for us.

What actually Greasemonkey is?

Greasemonkey is a Firefox add-on or plug-in which extends the functionality of the browser

by making changes as per the user's instructions written in user scripts. Greasemonkey can be used for adding new functions to web pages, fixing rendering bugs, combining data from multiple web pages, and numerous other purposes. These user scripts are written in JavaScript which is a lightweight programming language to add interactivity to HTML pages. It is obvious that to write your user script, you have to learn JavaScript first. This JavaScript manipulates the contents of a web page using the Document Object Model (DOM) interface and provides results that the user wants to see. Using the Document Object Model (DOM) interface means that the user scripts interacts with the objects within HTML, XHTML and XML documents. Objects or elements of DOM may be addressed and manipulated within the syntax of the JavaScript in user script. And this means that we have total control on the web page we see through Firefox if we have Greasemonkey installed.

Greasemonkey user scripts execute each time the page is loaded in browser and the user script repeatedly does this to make changes permanent. Some user scripts may be site specific which works only on a particular page. (for example Auto fill the username and password for Gmail as browser detects Gmail).

What can you do with Greasemonkey?

- Automatically fill forms on specific pages
- Alter the formatting of text, borders, graphics, etc on a page.
- Remove specific content, such as advertising, popups, even whole sections of a page.
- Alter content and layout beyond what the page author considered.
- Add links, buttons, or any other type of HTML element anywhere on the page.
- Enhance the content of the pages by retrieving correlating information from related pages on the same site, or other sites.
- Take advantage of extended JavaScript behavior to add previously non-existent functionality to pages.
- And lot more...

How does Greasemonkey power user scripts?

JavaScript, itself is a powerful programming extension to any HTML page. But Greasemonkey also has some good methods in its API which really enhances the power and functionality of any user script. The question is how we use it in our user scripts. Below I have listed some methods by task

Values

[GM_deleteValue](#)

This deletes a value from chrome that was previously set.

[GM_getValue](#)

A function intended to get stored values, see [GM_setValue](#) below.

[GM_listValues](#)

This API method retrieves an array of preference names that start with the branch's root.

[GM_setValue](#)

A function that accepts the name and value to store, persistently. This value can be retrieved later, even on a different invocation of the script, with [GM_getValue](#).

Resources

[GM_getResourceText](#)

Given a defined [resource](#), this method returns it as a string..

[GM_getResourceURL](#)

A function that loads an external resource via a URL, such as an image, and returns the string containing the base64 encoded in the data: URL scheme.

Common Task Helpers

[GM_addStyle](#)

A function, that takes one parameter, a string of CSS to apply to the page.

[GM_xmlHttpRequest](#)

A version of the XMLHttpRequest method underlying AJAX, this API makes arbitrary HTTP request to a page from other server or from same server to load or post data into current page.

[unsafeWindow](#)

This object provides access to the raw JavaScript `window` scope of the content page. It is most commonly used to access JavaScript variables on the page.

Others

GM_log

A function that accepts a parameter which will be routed to the Error Console, useful for examining values when writing a script.

GM_openInTab

Similar in spirit to `window.open()`, this function accepts a single parameter, the URL of a page to open in a new tab.

GM_registerMenuCommand

An advanced function which allows a user script to register a menu item, and command to run when clicked, in the Firefox user interface.

How to install?



You can see this little monkey on the right bottom corner of Firefox by installing Greasemonkey add-on from this link.

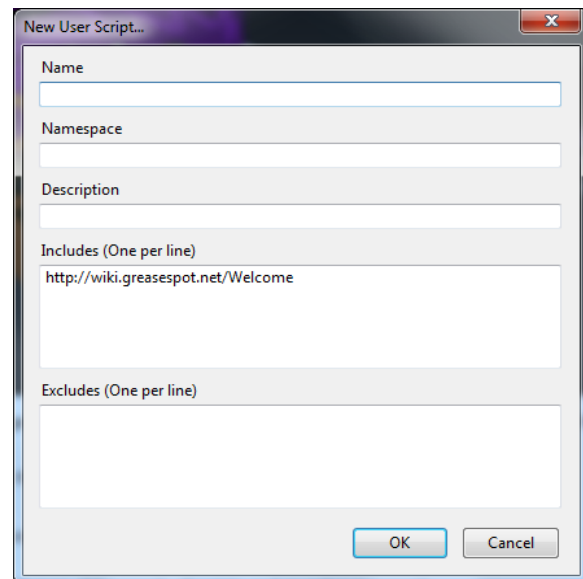
<https://addons.mozilla.org/en-US/firefox/addon/748>

1. Follow the above link and say “Add to Firefox”
2. Now a window will pop-up prompting user to install add-on to Firefox. Click Install Now and restart the browser as instructed.
3. Aright. Now that greasemonkey is installed, we check whether

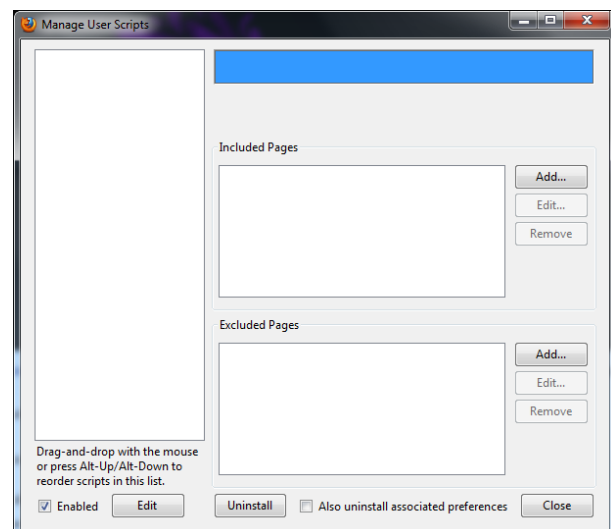
greasemonkey is enabled or not. To check it, in Firefox, go to Tools-> Greasemonkey-> Enable Here you can manage installed user scripts and add new user scripts.

How to use it?

1. You can add/import your own script



2. Also you can manage, enable, disable Greasemonkey, uninstall and edit user scripts. In above image you can see on the right side ‘Included Pages’ and ‘Excluded Pages’. You can add metadata entries for these two



options for selected Greasemonkey script. We will discuss description of metadata in the following points.

Now after adding it, the question is that what this monkey actually do. This monkey allows you to add your own scripts on every page load, means when your web page loads then your script does its work which is made for some specific purpose. Greasemonkey website says that 'Greasemonkey is a Firefox extension that allows you to customize the way web pages look and function.' Yes this is the function we actually need to personalize our browser, isn't it?

Resource

Now you have freedom to write your script add it to firefox and do just whatever you want. You have millions of free user scripts on the website <http://www.user script.com>.

Thousands of script developers have created user scripts to functionalize their browser and uploaded to user scripts.com. Some of them are useful for you or some can give you idea to create your own. Until now various users have written scripts for facebook page management, auto fill forms for particular site, to improve appearance of gmail, yahoo accounts, and also to hide advertisements from any site you visit.

Do it yourself

Now after you have used various user scripts from various users don't you want to try your hand on this exciting facility? Yes you will do this I am sure. Let me tell you there are lots tutorials, books written for the use of Greasemonkey. One of them is Greasemonkey Hacks which teaches you how

to write your own script, how to apply them for different purposes. About 100 scripts are described in this book along with source codes. But I recommend you to use your own mind, read some tutorials of JavaScript, take help from JavaScript to develop your own unique script. If you know HTML, JavaScript very well then go for it. You also have Google as the best internet teacher with you.

Supporting Browsers

When Greasemonkey was launched it was written only for firefox. But now browsers like Google Chrome and Mozilla based Flock are also supporting it. Flock and Firefox are directly supporting Greasemonkey by installing as an Add-On from tools menu. In case of Google Chrome Greasemonkey is not directly supported. In earlier versions right after 2.0 you have to trick the browser setting from installations files to get support of GreaseMonkey. Latest version of Google Chrome allows GreaseMonkey to execute as an extension to Chrome browser. Like Firefox we do not need GreaseMonkey Add-On to be installed to add user scripts. You just have to create your user script and simply drag it into the Chrome browser. Your user script will do actions when target or any page loads.

How to write User script?

1. Open any text editor you like
2. Write your user script to execute.
3. Save as file by adding extension “.user.js” after user script name.
e.g. If you wish to save user script with MyScript then file will look like this MyScript.user.js

Greasemonkey Metadata

As numerous user scripts exist, there is a provision to differentiate between user scripts installed on Greasemonkey Add-on. For this metadata is added above the code. This metadata tells Greasemonkey about the script itself, where it came from, and when to run it, version info, etc. As an example,

```
// ==User script==
// @name          Hello World
//               @namespace
http://clubhack.com/
// @description   example script to
alert "Hello world!" on every page
// @include      *
//               @exclude
http://clubhack.com/*
//               @exclude
http://www.clubhack.com /*
// ==/User script==
```

“//” is used to denote metadata line.

You can see metadata wrapper enclosed within

```
// ==User script==
//
// ==/User script==
```

Greasemonkey uses above tags to denote the start and end of your user script metadata. This section may be defined anywhere in your script, but it is usually declared on top of your code.

First item of metadata is

```
@name
```

This is the name of your user script. It will be displayed in the install dialog when you first install the script and later in the

“Manage User scripts” dialog. It should be as short as possible

@name is optional. If it is present in your user script, it may appear only once. If not present, it shows default name as filename of the user script, minus the .user.js extension.

Next is the namespace,

```
//               @namespace
http://clubhack.com/
```

This is a URL, and Greasemonkey uses it to distinguish user scripts that have the same name but are written by different authors. If you have a domain name, you can use it as your namespace. Otherwise you can use a default tag: URI.

@namespace is optional. If present in your user script, it may appear only once. If not present, it defaults to the domain from which the user downloaded the user script.

Next comes the description,

```
// @description   example script to
alert "Hello world!" on every page
```

This is a human-readable explanation of what the user script does. It is displayed in the install dialog when you first install the script, and later in the “Manage User scripts” dialog. It should be no more than two sentences.

@description is optional. If present in your user script, it may appear only once. If not present, it defaults to an empty string.

Don't forget to write the @description. Even if you are only writing user scripts for yourself, you will eventually end up with dozens of them, and administering them all in the “Manage User scripts” dialog will be much more difficult if you don't include a description. It will get easy to differentiate between many user scripts.

The next three lines are the most important items (from Greasemonkey's perspective): the @include and @exclude URLs.

```
// @include      *
//                                     @exclude
http://clubhack.com/*
//                                     @exclude
http://www.clubhack.com/*
```

These lines tell Greasemonkey on which sites you want your user script to execute. Both specify a URL, with the * character as a simple notation for part of the domain name or path. In this case, we are telling Greasemonkey to execute the Hello World script on all sites except <http://clubhack.com/> and <http://www.clubhack.com/>. Excludes take precedence over includes, so even though <http://clubhack.com/> matches * (all sites), it will be excluded because it also matches <http://clubhack.com/>*

@include and @exclude are optional. You may specify as many included and excluded URLs as you need, but you must specify each on its own line. If neither is specified, Greasemonkey will execute your user script on all sites (as if you had specified @include *).

User scripts Examples

1. Malware Script Detector

This is the very useful script while you are surfing on internet heavily and not conscious about which site is flagged susceptible to malicious infection by antivirus or Google. Web pages are built using HTML and enhanced using JavaScript. But some websites injects harmful scripts into computer via your browser and you do not get any traces of it. And after some time, the injected code starts to execute and do its designated

function. This user script is written to detect and alert when user script detects malicious JavaScript. It protects your computer from harmful attacks.

Click the link below to download/install this user script.

<http://userscripts.org/scripts/show/30284>

2. Post Interceptor

Intercept POST requests and let user modify before submit

Click the link below to download/install this user script.

<http://userscripts.org/scripts/show/743>

3. Web Page Finger Printer

For web2.0 security analysis. To be used with FireBug. For quick analysis, it provides the overall view of the current page contents - javascript, cookies, fuzzable links, form data. For security assessment, it provides recon scan, bruteforce scan, and fuzzing form. In what it differs from the thick-client full-fledged scanner is that this script is tied to the current url page and will not mess with the whole web site. Use it at your own risk. Feel free to send bugs.

Click the link below to download/install this user script.

<http://userscripts.org/scripts/show/30285>

4. Php Sec info Checker

Check phpinfo page for security and performance issues.

Click the link below to download/install this user script.

<http://userscripts.org/scripts/show/30287>

User script database

<http://www.userscript.org> is a website which has a database of a number of user scripts written by coders all over the world. All the user scripts are categorized by tags. So it gets easy to search. You can share your user script by registering to site and make your impact to Greasemonkey world by improving the web.

Be Aware...

Also I want to warn my readers... before installing any user script written by other coder, take a look at code first, check script data (Right click on script file, open with notepad) for trusted sources, check for code which may harm your computer by injecting malicious code (Malware). There may be a possibility that some scripts can keylog your browser data, and harm your privacy. You can read the material provided on Wikipedia, before you start using it. So be careful, be ethical with the tool I introduced to you.

Be Ethical...

As Greasemonkey allows us to write our own user script and allows us to take control of any web page we surf over the network, we should use this technology ethically to make use of Firefox efficiently. After reading this article some people may want to do nasty things which actually are not ethical in any way. But I expect from you all to use Greasemonkey for the purpose it was built and enjoy using it.



Sagar S Nangare

Sagar is IT Engineering Student. Currently working as a WEB DEVELOPER.

who is?

Whois Query under Microsoft

Introduction

WHOIS is a query and response protocol that provides information about the domains, networks and hosts.

If the domains found, the whois query would provide following information :-

1. Name, postal and email address of the person and contact numbers of the person under whose name the domain is registered.
2. Creation and Expiration Date.

Whois records are stored with the domain registrars.

How to carry out Whois query under Windows

Microsoft Windows Operating Systems does not have an inbuilt whois utility. However, you can download a small whois utility from Microsoft Technet (Windows Sysinternals) to carry out a whois search.

Download whois.zip from

<http://download.sysinternals.com/Files/Whois.zip>.

Unzip the file.

Open command prompt.

Navigate to the directory where you extracted the zip file.

Execute the command in the following format:

```
whois domainname [whois.server]
```

where domain name can be either a DNS name (e.g. www.abc.com) or IP address (e.g. 11.22.11.22)

For the whois.server, you have to select a server from the following whois server list.

[Whois Server List 1](#)

[Whois Server List 2](#)

An example - whois microsoft.com
whois.verisign-grs.com

```
C:\>whois microsoft.com whois.verisign-grs.com
```

Whois v1.01 - Domain information lookup utility

Sysinternals - www.sysinternals.com
Copyright (C) 2005 Mark Russinovich

Connecting to whois.verisign-grs.com...
Connecting to whois.verisign-grs.com...
Connecting to whois.melbourneit.com...

Domain Name..... microsoft.com
Creation Date..... 1991-05-02
Registration Date.... 2009-10-06
Expiry Date..... 2015-05-04
Organisation Name.... Microsoft Corporation
Organisation Address. One Microsoft Way
Organisation Address.
Organisation Address. Redmond
Organisation Address. 98052
Organisation Address. WA
Organisation Address. UNITED STATES

Admin Name..... Administrator .
Admin Address..... One Microsoft Way
Admin Address.....
Admin Address..... Redmond
Admin Address..... 98052
Admin Address..... WA
Admin Address..... UNITED STATES
Admin Email.....
domains@microsoft.com
Admin Phone..... +1.4258828080
Admin Fax.....

Tech Name..... Hostmaster .
Tech Address..... One Microsoft Way
Tech Address.....
Tech Address..... Redmond
Tech Address..... 98052
Tech Address..... WA
Tech Address..... UNITED STATES
Tech Email..... msnhst@microsoft.com
Tech Phone..... +1.4258828080
Tech Fax.....

Name Server..... NS2.MSFT.NET
Name Server..... NS4.MSFT.NET
Name Server..... NS1.MSFT.NET
Name Server..... NS5.MSFT.NET
Name Server..... NS3.MSFT.NET



Manu Zacharia

Microsoft MVP (Enterprise Security), ISLA-2010 (ISC)²
C|EH, C|HFI, MCP, CCNA, AFCEH
Certified ISO 27001:2005 (ISMS) Lead Auditor



Electronic Contracts (Part-1)

Contracts have become so common in daily life that most of the time we do not even realize that we have entered into one. Right from hiring an auto to buying airline tickets online, innumerable things in our daily lives are governed by contracts.

The **Indian Contract Act, 1872** governs the manner in which contracts are made and executed in India. It governs the way in which the provisions in a contract are implemented and codifies the effect of a breach of contractual provisions.

It provides a framework of rules and regulations which governs formation and performance of contract. The rights and duties of parties and terms of agreement are decided by the contracting parties themselves. The court of law acts to enforce agreement, in case of non performance.

Electronic contracts (contracts that are not paper based but rather in electronic form) are born out of the need for speed, convenience and efficiency.

Imagine a contract that an Indian exporter and an American importer wish to enter into. One option would be that one party first draws up two copies of the contract, signs them and courier them to the other,

who in turn signs both copies and couriers one copy back. The other option is that the two parties meet somewhere and sign the contract.

In the electronic age, the whole transaction can be completed in seconds, with both parties simply affixing their digital signatures to an electronic copy of the contract. There is no need for delayed couriers and additional travelling costs in such a scenario. There was initially an apprehension amongst the legislatures to recognize this modern technology, but now many countries have enacted laws to recognize electronic contracts. The conventional law relating to contracts is not sufficient to address all the issues that arise in electronic contracts.

As per the IT Act, 2000 only “Digital Signature” was the means for electronic authentication. This approach was not a technology neutral approach and the law was bound by a specific technology. The defect in this approach is that the law is bound by a specific technology, which in due course of time may be proven weak.

An example of this is the MD5 hash algorithm that at one time was considered suitable.

MD5 was prescribed as suitable by Rule 6 of the Information Technology (Certifying Authorities) Rules, 2000. MD5 was

subsequently proven weak by mathematicians.

In fact, Asian School of Cyber Laws had filed public interest litigation in the Bombay High Court on the same issue. Subsequently, the Information Technology (Certifying Authorities) Amendment Rules, 2009 amended the Rule 6 mentioned above.

MD5 was replaced by SHA-2.

The Information Technology (Amendment) Act, 2008 amends the technology dependent approach. It introduces the concept of **electronic signatures** in addition to digital signatures.

Electronic signatures is wider term covering **digital signatures, biometric authentication, etc**

It has a **technology neutral** approach and not bound by any specific technology.

Types of electronic signatures

- Based on the knowledge of the user or the recipient e.g. passwords, personal identification numbers (PINs)
- Those based on the physical features of the user (e.g. biometrics)
- Those based on the possession of an object by the user (e.g. codes or other

information stored on a magnetic card)

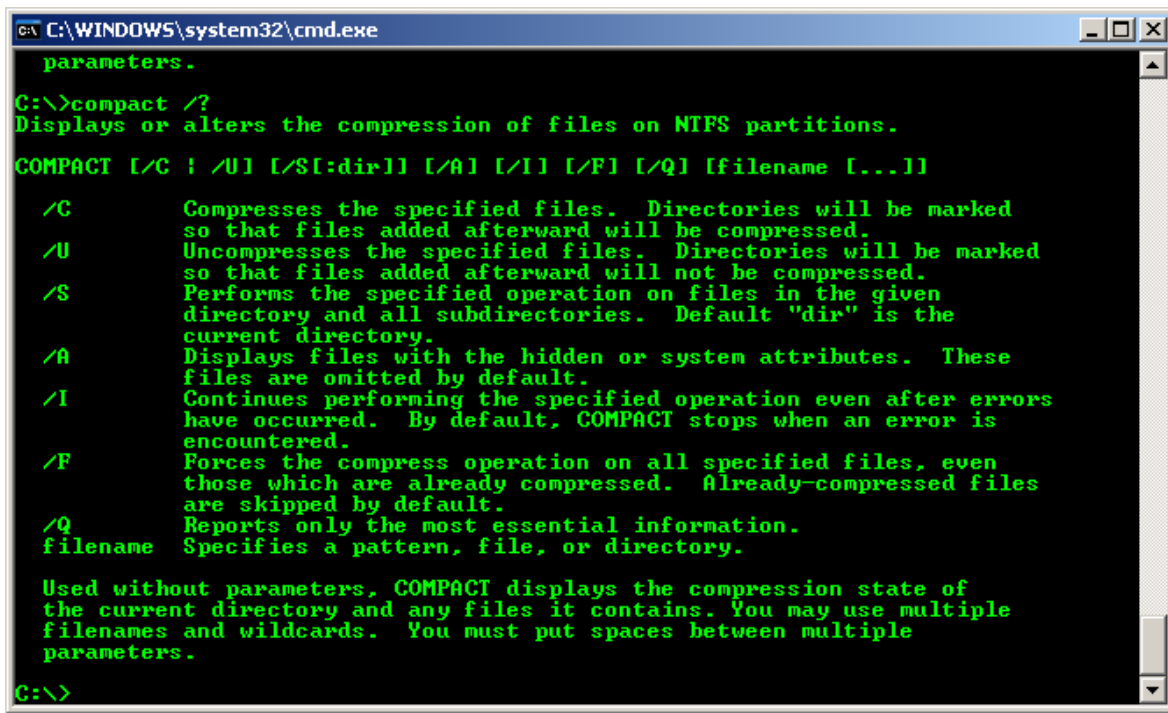
- Scanned handwritten signatures
- Signature by means of a digital pen
- Clickable “OK” or “I accept” boxes
- Digital signatures within a public key infrastructure (PKI)
- Biometric devices
- Hybrid solution like combined use of passwords and secure sockets layer (SSL)



Sagar Rahukar
sr@asianlaws.org

Sagar Rahukar, a Law graduate, is Head(Maharashtra) at Asian School of Cyber Laws. Sagar specializes in Cyber Law, Intellectual Property Law and Corporate Law. Sagar also teaches law at numerous educational institutes and has also trained officials from various law enforcement agencies.

Command LINE



```

C:\WINDOWS\system32\cmd.exe
parameters.
C:\>compact /?
Displays or alters the compression of files on NTFS partitions.
COMPACT [/C | /U] [/S[:dir]] [/A] [/I] [/F] [/Q] [filename [...]]

/C      Compresses the specified files. Directories will be marked
        so that files added afterward will be compressed.
/U      Uncompresses the specified files. Directories will be marked
        so that files added afterward will not be compressed.
/S      Performs the specified operation on files in the given
        directory and all subdirectories. Default "dir" is the
        current directory.
/A      Displays files with the hidden or system attributes. These
        files are omitted by default.
/I      Continues performing the specified operation even after errors
        have occurred. By default, COMPACT stops when an error is
        encountered.
/F      Forces the compress operation on all specified files, even
        those which are already compressed. Already-compressed files
        are skipped by default.
/Q      Reports only the most essential information.
filename Specifies a pattern, file, or directory.

Used without parameters, COMPACT displays the compression state of
the current directory and any files it contains. You may use multiple
filenames and wildcards. You must put spaces between multiple
parameters.
C:\>
  
```

Let's zip it up

Introduction

As we know Compression is the reduction in size of data to save space. For data compression ZIP format is used and it's in archive format. ZIP file contains one or more files that have been compressed to reduce file size, or stored as it is. This time we are going to reduce size of file or directory using command-line and save space on HDD.

WINDOWS

For compression of files or directories we are using COMPACT command in Windows. So let's play with 'COMPACT' in command-line.

"COMPACT" command is used for un-compressed NTFS files, it doesn't extract files from a compressed (zipped) file.

Let's see how to use this command.

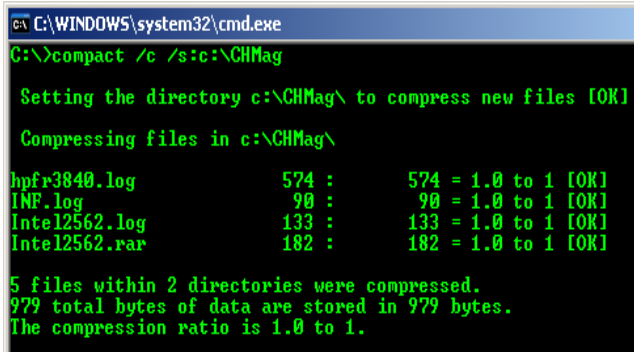
For getting help or all options use following command.

```
C:\>compact /?
```

For compressing the specified directory or file use following command

```
C:\>compact /c /s:c:\CHMag
```

/s- is used to all subdirectories of the specified directory. CHMag is my directory on C drive and it contains some files and subdirectories. After execution of this command CHMag folder gets compressed. For verification go to explorer and check that if the folder name is blue colored.



```

C:\WINDOWS\system32\cmd.exe
C:\>compact /c /s:c:\CHMag

Setting the directory c:\CHMag\ to compress new files [OK]

Compressing files in c:\CHMag\

hpfr3840.log      574 :      574 = 1.0 to 1 [OK]
INF.log          90 :      90 = 1.0 to 1 [OK]
Intel12562.log   133 :     133 = 1.0 to 1 [OK]
Intel12562.rar   182 :     182 = 1.0 to 1 [OK]

5 files within 2 directories were compressed.
979 total bytes of data are stored in 979 bytes.
The compression ratio is 1.0 to 1.

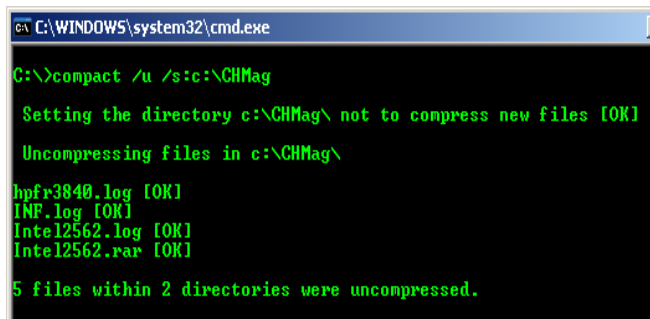
```

Figure 1

Fig.1 shows you detail of compresses the CHMag directory

For decompressing the specified directory or file use /u option i.e. for uncompressing the specified directory or file. (refer Fig.2)

```
C:\>compact /u /s:c:\CHMag
```



```

C:\WINDOWS\system32\cmd.exe
C:\>compact /u /s:c:\CHMag

Setting the directory c:\CHMag\ not to compress new files [OK]

Uncompressing files in c:\CHMag\

hpfr3840.log [OK]
INF.log [OK]
Intel12562.log [OK]
Intel12562.rar [OK]

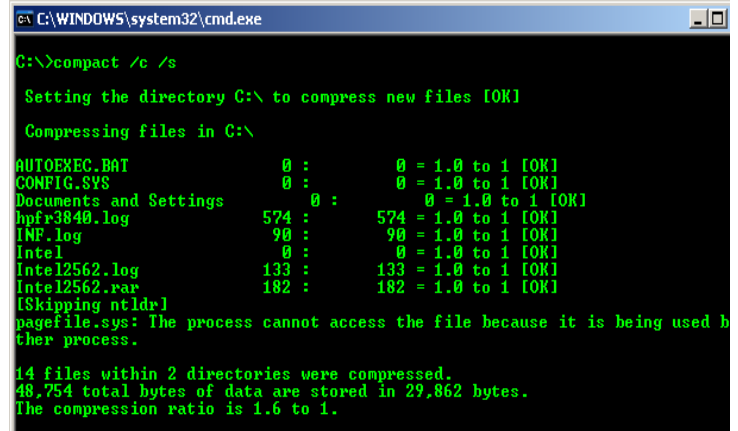
5 files within 2 directories were uncompressed.

```

Figure 2

To set the compression of the current directory, its subdirectories, and existing files use following command.

```
C:\>compact /c /s
```



```

C:\WINDOWS\system32\cmd.exe
C:\>compact /c /s

Setting the directory C:\ to compress new files [OK]

Compressing files in C:\

AUTOEXEC.BAT      0 :      0 = 1.0 to 1 [OK]
CONFIG.SYS        0 :      0 = 1.0 to 1 [OK]
Documents and Settings 0 :      0 = 1.0 to 1 [OK]
hpfr3840.log     574 :     574 = 1.0 to 1 [OK]
INF.log          90 :      90 = 1.0 to 1 [OK]
Intel            0 :      0 = 1.0 to 1 [OK]
Intel12562.log   133 :     133 = 1.0 to 1 [OK]
Intel12562.rar   182 :     182 = 1.0 to 1 [OK]
[Skipping ntldr]
pagefile.sys: The process cannot access the file because it is being used by
another process.

14 files within 2 directories were compressed.
48,754 total bytes of data are stored in 29,862 bytes.
The compression ratio is 1.6 to 1.

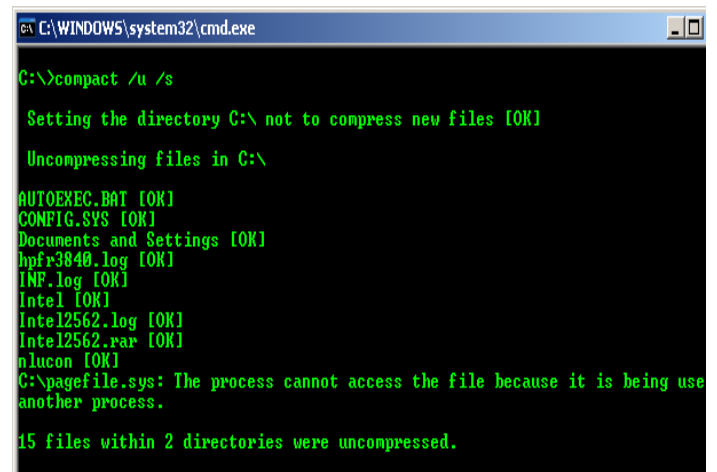
```

Figure 3

Fig.3 shows you compression of the C directory.

To Unset the compression of the current directory, its subdirectories, and existing files use the following command. (refer Fig.4)

```
C:\>compact /u /s
```



```

C:\WINDOWS\system32\cmd.exe
C:\>compact /u /s

Setting the directory C:\ not to compress new files [OK]

Uncompressing files in C:\

AUTOEXEC.BAT [OK]
CONFIG.SYS [OK]
Documents and Settings [OK]
hpfr3840.log [OK]
INF.log [OK]
Intel [OK]
Intel12562.log [OK]
Intel12562.rar [OK]
ntlucon [OK]
C:\pagefile.sys: The process cannot access the file because it is being use
another process.

15 files within 2 directories were uncompressed.

```

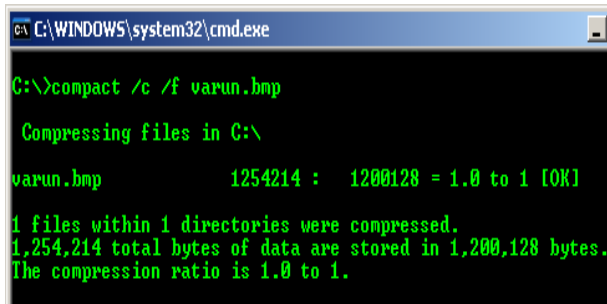
Figure 4

We can also compress the image files like jpg, png, bmp files using compact command

For compression of a .bmp file use following command

```
C:\>compact /c /f varun.bmp
```

“varun.bmp” is my file on C drive use your file name with the address. /f –used to force complete compression of the file. And to uncompress use /u option instead of /c (refer Fig.5)



```

C:\WINDOWS\system32\cmd.exe
C:\>compact /c /f varun.bmp
Compressing files in C:\
varun.bmp          1254214 :  1200128 = 1.0 to 1 [OK]
1 files within 1 directories were compressed.
1,254,214 total bytes of data are stored in 1,200,128 bytes.
The compression ratio is 1.0 to 1.
  
```

Figure 5

Linux

Now it's time to save disk space in Linux platform. We all know that Linux is open source and is enhanced by free advanced tools to operate on data. So in Linux platform we are going to use “**gzip**”.

gzip is also a standard internal command library in Linux used to reduce file size using Lempel-Zev compression algorithm. One important thing to remember about gzip is that it replaces your original file with a compressed version with .gz extension. The amount of compression varies with the type of data, but a typical text file will be reduced by 70 to 80 percent and other types accordingly vary

For Example, if you compress plain/text file clubhack which is 4.6 kb (4741 bytes). After compression it is reduced to

```
admin@clubhack:~$ gzip clubhack
```

After compressing using gzip, extension is changed like this

Clubhack.gz

.gz is added after file name which means file is compressed using gzip.

Now you have compressed file using gzip it's time to extract or decompress file. To do this enter command

```

admin@clubhack:~$ gunzip
clubhack.gz
                                or
admin@clubhack:~$ gzip -d
clubhack.gz
  
```

This command replaces this file back with original one 'clubhack' and extension .gz is removed.

Options:

-r : This is the most useful option which tells gzip and gunzip to recursively compress or decompress all files in the current directory and any subdirectories.

```
admin@clubhack:~$ gzip -r
somedirectory
```

Zip all files in a directory called somedirectory.

```
admin@clubhack:~$ gunzip -r
somedirectory
```

Unzip all files in the somedirectory.

-n : with this option you can tell gzip to use different levels of compression with the **n** flag, where **n** is a number from 1 to 9.

The **-1** flag means "fast but less efficient" compression, and **-9** means "slow but most efficient" compression. Values of **n** between 1 and 9 will vary speed and efficiency, and the default is **-6**. If you want to get the best possible compression and use the **-9** flag, like this:

```
admin@clubhack:~$ gzip -9
clubhack
```

-c : It's common to apply **gzip** to a tar file, which is why you see files with names like **clubhack.tar.gz** on Linux systems. When you want to extract the contents of a gzipped tar file, you have several choices. The first is to use **gunzip** followed by **tar**, like this:

```
admin@clubhack:~$ gunzip
clubhack.tar.gz

admin@clubhack-:~$ tar xvf
clubhack.tar
```

Or you can do it all in one command, like this:

```
admin@clubhack:~$ gunzip -c
clubhack.tar.gz | tar xvf -
```

The **-c** flag tells **gunzip** to decompress the file, but instead of creating a **clubhack.tar** file, it extracts the decompressed data directly to the **tar** command. The **tar** command on the right side of the pipeline looks a little strange, instead of a file name after the **xvf**, there's just a dash. The dash tells **tar** that the input is not an actual file on disk, but rather a stream of data from the pipeline.

Concatenating multiple files:

It is possible to concatenate multiple files in to one file. In that case **gunzip** will extract all files at once.

```
admin@clubhack:~$ gzip -c
file1 > clubhack.gz

admin@clubhack:~$ gzip -c
file2 >> clubhack.gz
```

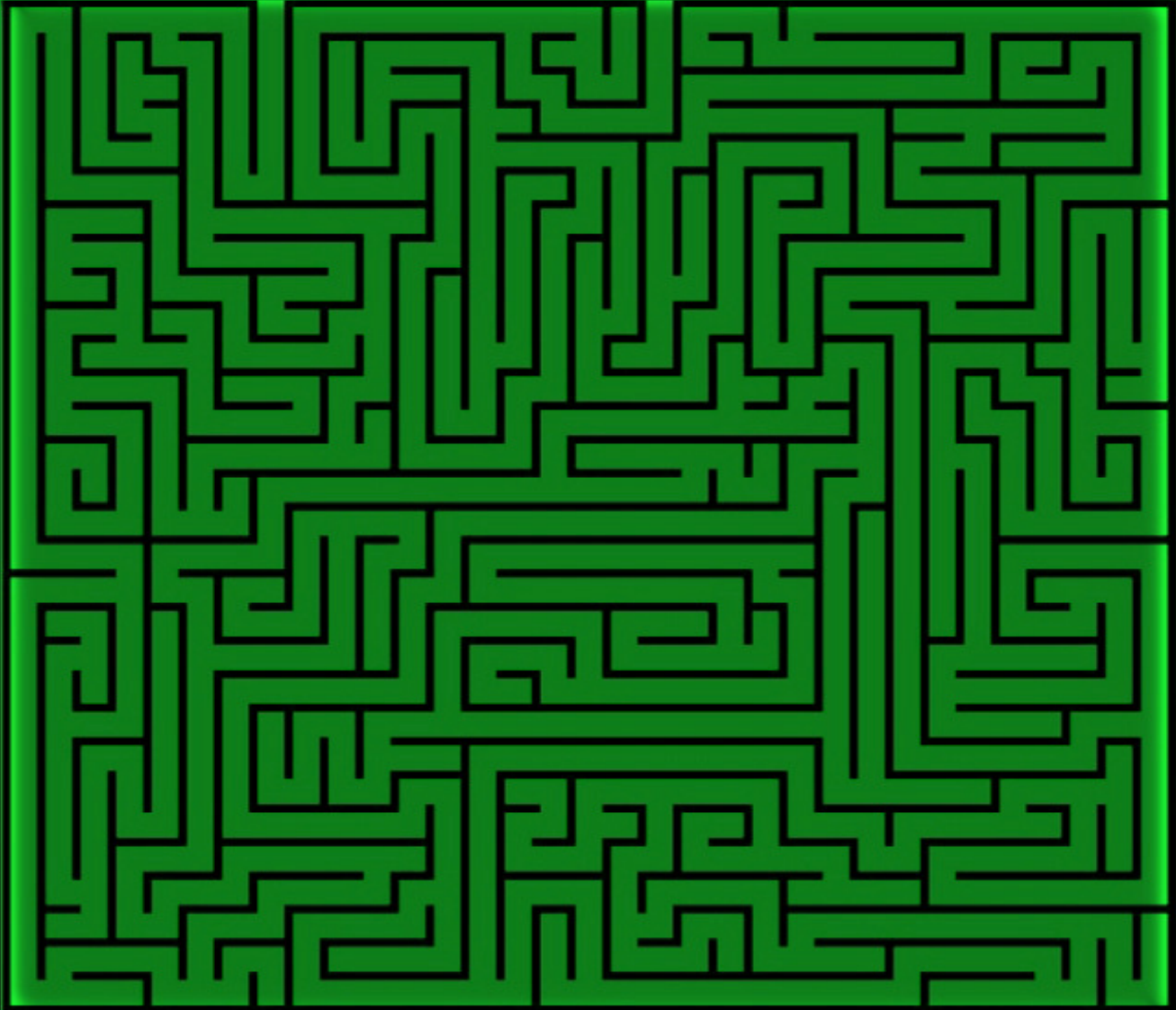
then to extract files

```
admin@clubhack:~$ gunzip -c
clubhack
```

Enjoy ☺



Varun Hirve
varun@chmag.in



@pankit_thakkar

It's amazing...What you can find, or lose on the network