

Forum:[Hack Forums](#)

Thread: [Shiru's Basic Security Guide \(5 chapters\)](#)

DeltaNox - July 14, 2011, 5:30 am

Shiru's Basic Security Guide

Index

- 1 Intro
- 2 Passwords
 - 2.1 Social Engineering
 - 2.2 Keyloggers and stealers
 - 2.3 Phishers
 - 2.4 Security questions
 - 2.5 Bruteforcing
 - 2.6 Building a strong password
- 3 Windows
 - 3.1 Anti-virus programs
 - 3.2 Avoiding Viruses
 - 3.3 Firewall
 - 3.4 Unnecessary features and services
 - 3.5 Tips
- 4 Linux
 - 4.1 The root account
 - 4.2 Viruses
 - 4.3 Firewall
 - 4.4 Tips
- 5 Webservers
 - 5.1 Brute-forcing
 - 5.2 DoS attacks
 - 5.3 Encryption
 - 5.4 SQL injection
 - 5.5 Javascript injection
 - 5.6 Tips

1 Intro

This relatively short guide covers information about some basic security features that are essential to any user that is connected to the internet. I'll explain some methods mainly to prevent other people from damaging or accessing your system without your authorization.

This guide is not even the tip of the iceberg and it's important to do the necessary research even if you know all these things. It might be a good idea to find info about the subject individually that's of interest to you. For most users running a desktop computer though, this guide should help you not getting your accounts hacked and to make sure you don't get infected by malicious software.

2 Passwords

Passwords are the most important part of preventing unwanted access to your system, that's why it's the first thing to discuss. They're the key to pretty much everything. It's like the key to your house; no matter how complex and secure your lock is, if you have the key you can get in.

To prevent people from finding out your password you first need to know how they do it. There are numerous methods of which the most popular will be discussed here.

2.1 Social Engineering

Social engineering is a method of manipulating a person into giving out vital information that will grant them access to whatever it is they're looking for. The solution to not getting tricked is simply to never ever give your password to anyone. No matter who is asking.

An example would be you receiving an email from a website your are registered to asking for your login details. They'll tell you things like your account is going to get deleted if you don't reply. Remember that no organisation or website will ever ask you for you password. If it happens it's always someone trying to get your password.

If you're still not sure you should contact the organisation itself, though never reply to the email itself.

2.2 Keyloggers and stealers

Keylogging is when someone succeeds in running a program on your computer that logs every key you press and sends the information to that person. Passwords and other data you've used to log in are often stored in your computer, e.g. when you check the "remember password" on a site that requires you to login. Stealers "steal" the information that is stored there.

Most of the time social engineering is used to make you download these programs. Since keyloggers and stealers can be categorised as malicious programs, check the Anti-virus part under the Windows section for information on how to prevent such programs.

2.3 Phishers

Phishers are programs or websites that appear legit but in reality send your login info to a person with malicious intend. When you log in to a website, after following a link always check the url in the addressbar to see if it's the actual website you want to visit.

2.4 Security questions

Some websites have security questions which are used in case you've forgotten your password. They're somewhat of a second password to allow a person to reset his other password. Which in my opinion doesn't make much sense because it's a serious flaw in having a secure account.

A cracker might be able to gain access to your account simply by filling in security questions such as "What is your mom's name?". The answers can easily be social engineerd since it's much less suspicious than asking for a password. And because of social networking, most of this information can easily be found online. To prevent this, answer the question in a way that doesn't make any sense such as your birthplace being Jupiter or the 4th dimension.

2.5 Bruteforcing

There's more to making a strong password than creating a password that can't be guessed easily, which brings me to bruteforcing. Bruteforcing is running a tool that attempts to connect trough every possible password combination there is. Theoretically it will eventually find your password no matter what. At this point though(this might change in the future), if you have a good password, it will take long to crack a password, and by long I mean very long, e.g: A password with 7 characters, using punctuation, digits, lower and uppercase characters being attacked at 500 passwords per second can take up to 10 months to crack.

In case of personal emails or sites like facebook this is unlikely to happen as no one is going to run a computer for that long just to mess with your account. When it comes to servers though, there are

a number of precautions you should be taking. These will be discussed in the web servers chapter, as I think it's more relevant there.

[b]2.6 Building a strong password[/b]

To build a password that can't be cracked easily you can use the following guidelines:

- Always use 1 special character in your password such as: "=", "@", "&", "\$,..."
- Use numbers, upper and lowercase characters
- Make the password at least 7 characters
- Never make the password into something that relates to you, like your phone number, address or favourite video game.

These are a few links to secure password generators:

- <http://www.pctools.com/guides/password>
- <https://addons.mozilla.org/en-us/firefox/addon/securepassword-generator>
- <http://www.safepasswd.com/>

3 Windows

The first OS we'll discuss is windows since it's the most popular operating system.

As in most situations, the best protection against users with malicious intent remains common sense. Though there are certain things you might want to consider for protection nonetheless.

3.1 Anti-virus programs

Since most malicious programs are specifically designed to attack windows users, you'll probably want to install an anti-virus. First of all, there are a bunch of AV's you'd best avoid when getting a decent protection.

The ones to avoid are:

- Avast
- AVG
- Avira
- Bitdefender
- Kaspersky
- Spybot S&D
- McAfee

The reason these are listed is because they have a high resource usage and have a relatively low detection rate. Also they are commercial and not always free. I'm not saying you shouldn't be using these programs, but for a good AV you might want to consider the following.

Advised Anti-virus programs:

- Malwarebytes
- Microsoft Security Essentials
- Nod32

3.2 Avoiding Viruses

Even though there are tools to prevent you downloading a virus, not all viruses are detected. Just because your PC has a decent protection doesn't mean you can go around and download anything carelessly without facing the consequences eventually. The most important thing should be obvious: know what you're downloading. If there's any doubt at all or whenever you don't know if the source is reliable, either google the name of the file you're going to run (most of the time this will be an executable file) or use an online virus scanner or preferably both.

Online virusscanners will analyze the file and display significant information of the program based on the detection rates of many AV's.

A few good online virusscanners:

- <http://vscan.novirusthanks.org/>
- <http://www.virustotal.com/>
- <http://virusscan.jotti.org/en>
- <http://anubis.iseclab.org/?Anubis>

After scanning your file on one of these, most of the time the file will be clean. It's still possible though for the application to be infected, e.g. if the file has been crypted. This means the file's "signature" has been changed so an antivirus won't recognize it's true intent.

In case you're still not sure about running a file you can test to see what happens using sandboxie. The website's description: "Sandboxie runs your programs in an isolated space which prevents them from making permanent changes to other programs and data in your computer." This means you can run a program and see what happens without it doing any damage if it's a virus.

Sandboxie website:

- <http://www.sandboxie.com/>

3.3 Firewall

A firewall is not as much used as protection against malicious executables but rather to protect you from what would happen once such a program has been able access your system, most of the time this being a trojan horse. What it does is prevent people or programs connecting to your computer that shouldn't be. Connections are made through ports, in a firewall you can define which ports should be open and which shouldn't, this will prevent external connections that aren't authorized. Keep in mind though, that attackers will often scan your ip in order to find out which exact ports are open and then use those ports to access your system.

You can also disallow an ip-address, for example when it attempts to login a specified number of times, block the ip of where the attempts are coming from.

A list of good firewalls:

- Comodo Firewall Pro
- ESET Smart Security
- ZoneAlarm Free Firewall
- Windows Firewall

3.4 Unnecessary features and services

When not running a server(which will be discussed individually later) there are a lot of programs and services you won't really need and might even be hazardous to your security such as remote access. I'm not going to discuss these in detail but if you want more info google's your friend.

- IIS – Microsoft's Internet Information Services
- NetMeeting Remote Desktop Sharing
- Remote Desktop Help Session Manager
- Remote Registry
- Routing and Remote Access
- Simple File Sharing
- SSDP Discovery Service
- Telnet

3.5 Tips:

- Update regularly, especially your AV programs.
- Regularly remove your internet history, cookies and any other things stored by your browser.
- Don't use internet explorer(any other browser will do)
- Make sure Windows is really the operating system you want for your specific needs.
- Use a password to login
- If you're using wireless make sure it's secure and has a good password.

4. Linux

Linux is different than windows when it comes to security, but not necessarily safer.

I tried finding information online about which one is more secure and stumbled upon various articles, one that told me "Why Linux is more secure than windows.", and another that said "Windows is more secure than Linux."

A lot of the time both Linux and Windows users will argue that their system is the most secure one, but personally I think it's a matter of making the right choices: "What am I going to use for what I want to get done?". When you're going to do something, do your research on how to do it, in this case selecting the right OS/kernel

4.1 The root account

The root user also known as the superuser, is the account that has priviledges to do anything. Basically you need to prevent anyone that isn't you from accessing the root (it's best if you don't use it yourself too often either) since as a superuser a simple "rm -rf" command is enough to wipe everything on your system before you'll know it.

To prevent unwanted root access use a very strong password of at least 15 characters(check the password chapter for more about passwords).

4.2 Viruses

Even though malware is generally not common on a linux/unix system they do exists. That's also why you'll want to prevent executing files as root. At the moment though, these are not really a threat and considering other security measures you shouldn't need to worry too much about it.

You might want to read up about it nonetheless in case your system somehow gets infected anyway.

Some links:

- http://en.wikipedia.org/wiki/Linux_malware
- <http://www.linux.com/archive/feature/128450>

-

http://www.pcworld.com/businesscenter/article/198686/linux_trojan_raises_malware_concerns.html

4.3 Firewall

The Linux kernel has a powerful firewall. The root user can configure this firewall trough an application called iptables. For more information about what a firewall is check the windows chapter.

Iptables reads certain rules that are placed into tables, either INPUT, OUTPUT or FORWARD. According to these rules the firewall will either drop or allow packets that are incoming(INPUT), outgoing(OUTPUT) or passing trough(FORWARD).

The root user can specify these rules. It's always best to disallow all connections except those you specifically need.

For information about what everything means:

- <https://help.ubuntu.com/community/IptablesHowTo>

4.4 Tips

- When connecting remotely use SSH, not telnet. Data sent by telnet is not encrypted and can be read by people on the same network by using a packet sniffer.
- Update regularly: `apt-get update` && `apt-get upgrade`
- Anything that doesn't require root shouldn't be done through the root account
- remove packages you don't need, to list all installed packages: `dpkg --get-selections`

Other links:

- <http://www.faqsg.org/docs/gazette/tips.html>

- <http://www.cyberciti.biz/tips/linux-security.html>

5 Webservers

Most of the previous rules also apply to a webserver such as passwords and firewall settings though web servers require additional things to consider.

5.1 Bruteforcing

See the passwords chapter for information about what bruteforcing is. Preventing bruteforcing will be discussed in this chapter.

If your website has a login form, one of the first things you should do is to restrict the number of times a user can attempt log in with incorrect details, then disallow that user to log in for a certain amount of time. To do this you can keep a variable for each user. Once that variable has reached the maximum allowed login attempts, you disallow the user to log in. Once the time limited has been reached you can reset that variable to 0.

Keep in mind though that this can be used as an exploit to block the user's account by sending constant login requests.

Other options are:

- Giving a user the ability to allow login only from specified IP addresses.
- Captcha: <http://www.captcha.net/>

For attacks against the server there are various tools that can assist you into preventing a user from attempting to log in a number of times. BFD is a great tool for this:

<http://www.rfxn.com/projects/brute-force-detection>

There are a lot of methods to prevent these types of attacks. Which ones and how many you're going to use depends on how secure you need your server to be.

A few useful links:

- https://www.owasp.org/index.php/Blocking_Brute_Force_Attacks

- <http://www.ubuntugeek.com/how-to-setup-logwatch-on-ubuntu-desktopserver.html>

- <http://www.rfxn.com/projects>

- http://kevin.vanzonneveld.net/techblog/article/block_brute_force_attacks_with_ip_tables

5.2 DoS attacks

There can be various causes to a DoS attack. DoS means Denial of Service. This happens when a server receives more data than it can actually handle and slows down or even crashes. There are various types of DoS attacks and depending on what specifically might cause this you'll need to take the proper precautions.

To protect a website from DoSing your sever proper settings(or coding) are neccesary, such as setting a time limit between changes that can be made by other users like posts in forums.

There's also the type of attack that's specifically intended to take down your server for whatever reason by sending large amounts of packages. A lot of the time these attacks come from botnets. Botnets are collections of compromised computers(often by malicious software). This type of attack is know as a Distributed DoS attack or DDoS attack. These are often difficult to deal with or even impossible to stop completely.

Preventing such an attack requires proper software configuration and decent hardware. Since there are a lot types of attacks and how to prevent them; check the following links for more information.

Links:

- <http://learn-networking.com/network-security/how-to-prevent-denial-of-service-attacks>
- <http://www.linuxsecurity.com/content/view/121960/49/>
- <http://www.linuxjournal.com/article/6749>
- <http://technet.microsoft.com/en-us/library/cc750213.aspx>

5.3 Encryption

When encrypting something it changes whatever is encrypted into a series of apparant meaningless characters. In reality it's actually a word changed by a number of rules. I'll give an example.

If the rules for encryption are these:

Code t = a

e = 2

x = band we run the word "text" trough a program that encrypts it based on this set of rules we get "a2ba". So let's say the password to a certain account is "text". When that account is created and stored in a database the password should first be encrypted so it's stored in the database as "a2ba". Whenever the user tries to log in the password he enters will be encrypted. Only if that encrypted password equals the one in the database he should be allowed in.

This is useful to protect sensitive information in your database from being read directly. If one succeeds in hacking the database he will still have to decrypt that information. To decrypt something an attacker will also often resort to bruteforcing. You can try to write your own algorithm though most databases apply more commonly used encryption methods such as md5.

A few links:

- <http://www.cs.auckland.ac.nz/~pgut001/tutorial/>
- <http://www.cryptographyworld.com/>
- http://en.wikipedia.org/wiki/List_of_algorithms

5.4 SQL injection

SQL injection is simply as it says, injecting SQL code into an inputfield or the url or you website. This allows the user to access your database, e.g. if doing a select to get passwords: "SELECT * FROM `tblPasswords`". This will result in a list of all the passwords in your table(if that table exists of course).

To prevent SQL injection you need to properly check the values in your code before sending them to the database. In php there's a function for that:

PHP Code

```
<?php
function sql_quote( $value )
{
```

```

if( get_magic_quotes_gpc() )
{
    $value = stripslashes( $value );
}
//check if this function exists
if( function_exists( "mysql_real_escape_string" ) )
{
    $value = mysql_real_escape_string( $value );
}
//for PHP version < 4.3.0 use addslashes
else
{
    $value = addslashes( $value );
}
return $value;
}

```

After adding this function when doing a post you can use it like this

PHP Code

```
<?php
```

```
if (isset($_POST['val'])) $val = sql_quote($_POST['val']);
```

In asp.net you need to use stored procedures:

PHP Code

```
<?php
```

```

SqlDataAdapter myCommand = new SqlDataAdapter("login", conn);
myCommand.SelectCommand.CommandType = CommandType.StoredProcedure;
SqlParameter parm = myCommand.SelectCommand.Parameters.Add("@au_id",
    SqlDbType.VarChar, 11);
parm.Value = Login.Text;

```

Also don't forget to check the input before sending it the database so it actually has the correct format and such.

5.5 Javascript Injection

Javascript can be used to manipulate values in forms and change cookie settings. Unlike php, javascript can be edited to whatever the user wants and this could result in security issues if you're not careful. For example a user can change cookie parameters by using javascript like this:

Code javascript:void(document.cookie="parametername=somevalue");

To change the values of a textbox, or anything else for that matter such as maximum input length, textbox type, etc..

Code javascript:void(document.forms[0].someElement.value="somevalue"); It's also possible to change things by simply using development tools like firebug.

This allows the user to insert values into the database by doing a submit.

To prevent this, use your server side language and sql database to validate and format input values correctly. NEVER rely on client side validation.

5.6 Tips

- Set mysql to disallow admin login from remote systems
- Limit max processes per user
- Use SSH to access a remote server
- Make sure you have the latest stable version of your software installed

iStrip - July 14, 2011, 6:41 am

Nice guide and all, but I'm pretty sure Malware-bytes isn't an Anti-Virus and explain, how does Avira have a low detection rate?

Eminent - July 14, 2011, 11:58 am

Dang, I'm surprised all that fit on one post.

5 starred.

The Archenemy - July 14, 2011, 1:24 pm

Quote:The one's to avoid are:

- Avast
- AVG
- Avira
- Bitdefender
- Kaspersky
- Spybot S&D
- McAfee

Advised Anti-virus programs:

- Malwarebytes
- Microsoft Security Essentials
- Nod32

Excuse me?

>MSE is rubbish for anyone who actually uses the internet for more than 30min. at a time.

>MBAM is NOT an AV, its a scanner which compliments your AV.

>Spybot S&D the same rules as MBAM

>You made a typo and put "the ones to avoid" rather than "the ones you should use"

>Mcafee and AVG are okay, not the best IMO.

Chutiya hun mien - July 14, 2011, 2:11 pm

Very nice and useful post bro.. I have bookmarked this. Please do not delete it. Thanks :D

Ui Hirasawa - July 14, 2011, 2:11 pm

To be honest, the Anti Viruses you said were good are not good at all.

The one's to avoid are:

- Avast
- AVG
- Avira
- Bitdefender
- Kaspersky
- Spybot S&D
- McAfee

--

Here I'll fix it for you.

Advised Anti-virus programs:

- Avira
- Avast

The ones to Avoid

- Every other one you listed.

shravansofts - *July 21, 2011, 4:48 am*

Thanks for the Info!

keep them posting

RMustang - *July 22, 2011, 12:31 am*

This thread is really good and all, but Avira isn't really a resource hog, Microsoft Security Essentials is the second biggest resource hog right after Kaspersky. Also, VirusTotal.com isn't really a recommended website to use for files, unless you think they are infected and want the file to be distributed. The rest is fine, good job ?

DeltaNox - *August 12, 2011, 12:33 am*

Thanks for the comments. The info about the AV's I had to look up cause I don't use an AV so that's why I've made some mistakes there. I'd update it, but it's two posts in a row so I can't edit it without having to delete half the post or post the second part under the comments.