# Malware

Malware is any kind of software designed to be harmful. A virus is one example of a type of malware, but there are many others.

To do anything, the malware must get a chance to run on a system in the first place; getting onto the system is called *infection*. Malware has a *payload* – what the malware actually does on a computer system. The people who create and send out the malware also have a *goal*. The payload of the malware may achieve the goal, or only contribute to it.

Many types of malware need to stay on a system for a long time, so they may do some form of *self-preservation* – actions that keep them on the system and make them hard to remove.

Many types of malware *self-replicate*, that is, the malware program makes another, separate copy of itself. This may be part of how malware does infection, how it does self-preservation, or its payload.

## Antivirus

Programs that protect systems from malware are often called antivirus programs, even though they usually protect from more types of malware than just viruses. Their job is to make infection harder, and to detect and remove malware once it is in the system.

But how can they detect malware? Many people think that antivirus programs somehow watch for "bad things" to happen. It would be wonderful to live in a world where programs had a magical badness detection system! But we don't know how to do that. Any type of malware is just a program, made of the same

instructions in the same machine language as any other program. There are *very* few actions that antivirus software can detect and know for sure that a program is malware doing something harmful rather than a normal program working as intended.

So, instead of watching for "bad things," antivirus programs detect individual malware by recognizing them specifically. A *definition* or *signature* is part of a program – a specific sequence of instructions – which only that program has, and no other program has. Antivirus programs keep huge lists of these definitions and constantly check files on the computer and programs running in Main Memory against them.

Essentially, antivirus programs are using a wanted poster system, checking constantly against all the pictures of known "criminals" they have. For this to work, we need to discover the criminals in the first place. Companies that make antivirus programs have labs full of computers running programs that automatically visit web pages and download files to try to get infected by new malware. Since that is the only purpose of these computers, any new program that runs on them is suspected malware and gets checked. When they get infected with something they don't already have a signature for, the company can manually observe what the malware does and find its signature. Then they can provide that signature to customers so that particular malware becomes recognizable.

This is why many antivirus programs are free, but the regular updates to definitions cost a subscription fee – if you don't have the latest definitions, your antivirus can't detect recent malware. And if your computer happens to be infected before your antivirus company detects and finds the signature for the malware, then your antivirus program won't be able to detect it – if a criminal

comes by before you get the wanted poster, you won't recognize the danger in time.

There are many antivirus programs available, and you should definitely have one running on your machine and updating its definitions (usually daily, or even more often) at all times. Many antivirus companies offer freeware licenses for personal use, and these can offer protection as good as high-cost commercial packages. Certainly a free program with up-to-date definitions is better protection than an expensive program that hasn't been getting updates!

# Infection

Malware needs to get onto a system to run at least once to achieve its payload. This infection is generally achieved through either software exploits or social engineering; that is, by taking advantage of a known weakness in how a program works, or a known weakness of how humans behave.

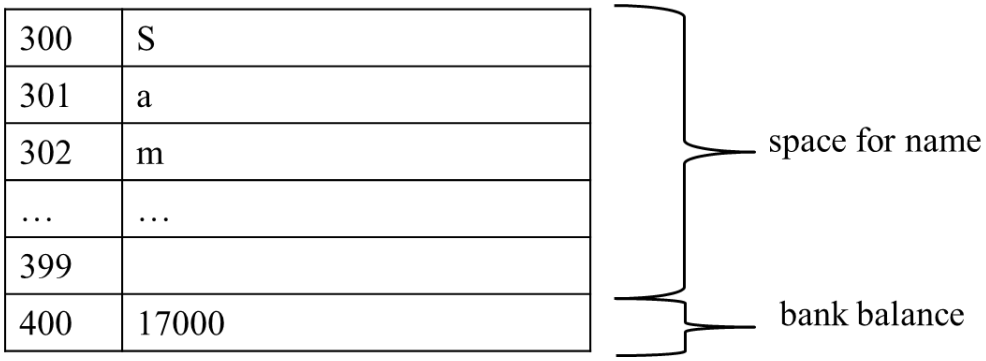
## Software Exploit

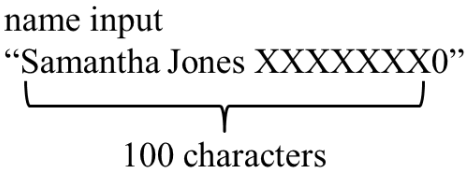
Programs often have features that work properly if users behave as the programmers expect, but can cause different and harmful behavior if deliberately abused. Taking advantage of such a weakness in the software to act against the user's wishes is a *software exploit*.

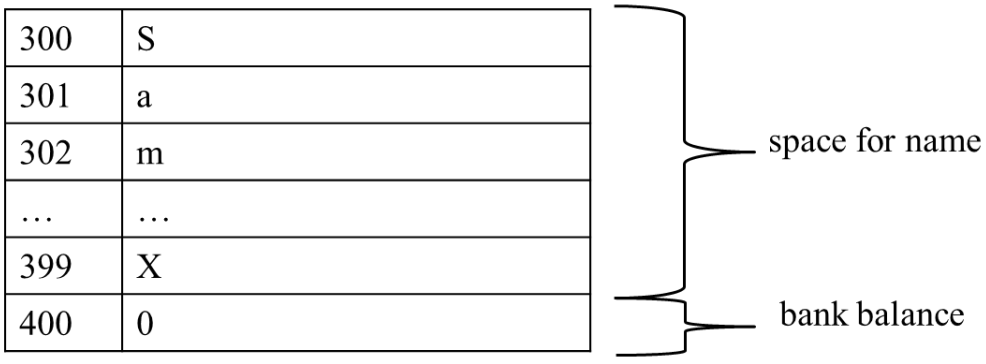
One common type of software exploit is *buffer overflow*. Here is a simplified example: Suppose we have a banking program. Some of the data the program has to store is the customer's name and their bank balance. The programmer set aside enough space in MM for 100 letters for the name, thinking nobody's name is longer than that, so they can accept anything put in for the

name and it will fit. Immediately after the space for the name in MM is the space for the balance, which the programmer protects so that it can't be changed easily.

If someone figured out that this was how the program worked, they could enter a name that was 100 letters followed by a number. The letters would fill up the space set aside for the name (the buffer) and the part left over, the number, would overflow this and go into the next space, the one for the bank balance. This would allow someone to set the bank balance to anything they wanted just by setting the name.




Once this exploit is known, the programmer (or their replacement!) can change things so that the program simply does

not accept data past the first 100 letters of the name entered by the user (which they should have done in the first place). Then the bad guys will start looking for a new weakness in the program.

This particular buffer overflow example would not get malware onto the system. But imagine a similar situation where the space after the buffer had program instructions instead of more data in it. This would allow someone to put the malware instructions into part of MM where they would later be run, which is all the malware needs.

Software exploits in web browsers are a very common way for malware to infect personal computers; in some cases it is enough to visit a website, and in the process of showing you the page, the browser's weakness allows the malware onto your system. This could be achieved by just one ad shown on a page, meaning that well known, legitimate sites can be involved in distributing malware without knowing it.

There are many types of weakness in software that could be turned into a software exploit, but programmers are becoming better at avoiding creating these in the first place.

## Social Engineering

If there is no weakness in the software, the other common approach to infection is to take advantage of common ways humans behave – *social engineering*. The user is tricked into doing something that allows the malware onto their system.

As an example: when someone downloads a new free program, it may have a malware program inside it or bundled with it (this might even be explained in the EULA, making it possibly legal!). This is taking advantage of people liking free things. Very

quickly, people came to mistrust free software because of this approach, even though a *very* small percentage of freeware ever actually had malware attached.

Instead, malware now often preys on people's fear of ... malware! A website flashes up a warning that your computer may be in danger and offers a free check. People are nervous about malware and accept this check, voluntarily downloading what turns out to be... malware.

Many email programs used to automatically open files attached to emails, which led to a lot of malware being run automatically. When this software exploit was (mostly) eliminated, the approach was updated to social engineering. Text in the body of the email is now written to imply that the attachment is a picture of something cute, a work-related document you have been waiting for, or official paperwork you need to read. This is often enough to trick people into opening the attached file, which allows the malware to run.

## Self-Preservation and Self-Replication

Many types of malware need to stay on a system past the initial infection, either to continuously run their payload, or to run it at some later date, so they need to do some form of self-preservation—working to stay on the system and not be removed.

The most effective form of self-preservation is to hide. If the users of the system do not notice anything is wrong, and there is no antivirus (or it is not updated with the definition for this malware) then the malware has no trouble staying on the system.

Some forms of malware cannot hide that they are there because their payload is too obvious. In this case there are various tricks they can adopt. The malware program's file may move from directory to directory on the hard drive, making it difficult to remove. Or it may make many copies of itself – self-replication—often faster than the copies can be deleted. It may hide itself in files with the same name as existing files on the system.

To avoid detection by antivirus programs, some malware *mutates*; it makes a copy of itself with minor changes to its instructions, for instance changing the order slightly or adding instructions that have no effect, so that it still does the same thing but no longer matches the known definition.

More sophisticated malware may attack antivirus software itself, if an exploit in the software can be found.

## Payload and Goal

The goals of people who create and distribute malware are generally profit, attention, or malice. The payload of the malware may achieve the goal, but often only supports it.

One type of payload of malware is to damage a system, usually by destroying files. Sometimes instead the system is temporarily disrupted, meaning that it cannot work or communicate with other systems. If this leads to users being unable to use some service, such as a website or email, this is called a *denial of service*.

If the goal is profit, malware's payload is often to steal information, which supports the overall goal of stealing money. If, for instance, the malware can collect your banking id and

password, the creator of the malware can then use this to take your money.

## Other Computer Threats

With the rise of interest in social engineering as a way to get malware onto systems, many criminals realized that they didn't in fact need the malware at all; it was enough to trick people into actions that furthered some goal.

Social engineering can be used against users even when there is no malware involved at all. For example, p*hishing* emails try to get information from people by pretending to be someone else, usually an official organization. This is just a text email; but the way it is written tries to trick the user into doing something.

For instance, you might receive an email that looks as if it comes from your bank, and says they have had a security issue. They ask you to mail back your account number and web banking name and password. In fact, no legitimate organization should ever ask for passwords to be sent to them, but if you did, that would get the information without their needing any malware at all.

A more sophisticated example of phishing might instead provide a link and ask you to log in. The link goes to a page that looks like your bank, but the page is a fake, and the information you enter is sent back to someone who will steal your money. This isn't malware, just a fake web page.

Many forms of identity theft – collecting enough information about you to pretend to *be* you, either to get access to your existing resources or to create new accounts in your name –

similarly require no malware, just tricking you into providing the information. For instance, suppose you see a quiz going around on a social media site that asks you to post your "superhero name" based on the name of your first pet and the street you grew up on. Seems silly but harmless... unless you have used that information to answer security questions for some of your accounts. Many security thieves have found they can get enough information to break into someone's account just by looking at the information they have already voluntarily posted online.

It may not seem particularly useful for someone to get into your HCC email or social networking account, but once they can do that, they can use this as leverage to get more information, for instance pretending to be you in order to have another organization give them *more* information about you. In general, for all accounts, choose some information that you will only ever use for security, and make sure that none of this is information you will ever post about socially.

## Viruses

Viruses are the most well-known type of malware, although they are no longer the most widespread.

The most common payload for viruses is to destroy data on the victim computer. A *time bomb* performs its payload at a certain time, often on a meaningful date such as New Year's or Valentine's Day.

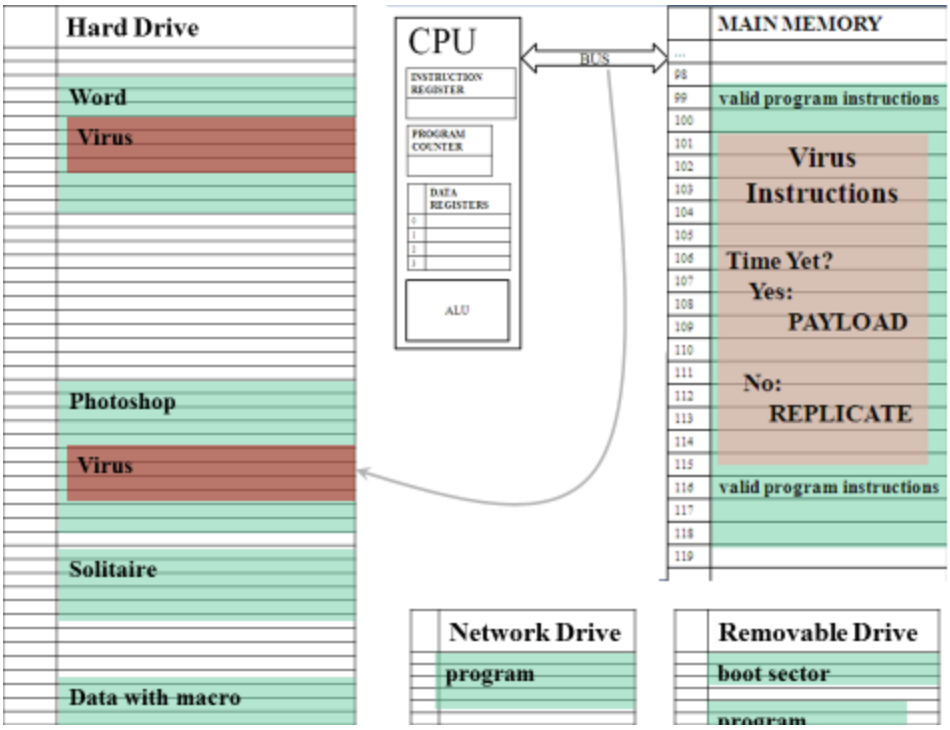
A *logic bomb* performs its payload in response to some other event on the computer, such as the user starting a certain program or opening a certain file. Although logic bombs are popular in fiction, most real viruses are time bombs, because the goal of most viruses is to get attention, such as news stories; if

everyone’s computer loses data on the same day, that is much more likely to be widely reported.

The virus is a program that has instructions to achieve its payload. Like any other program, these instructions must be in main memory to run, and will only be fetched if the program counter is eventually set to their addresses. Also, it must only do the payload if it is the correct date (for a time bomb, or if the chosen event has occurred, for a logic bomb), so it must contain instructions that check whether it is time to run the payload.

So, the virus needs to be in a set of instructions that are likely to be run on the chosen day. Generally, users are only willing to run their own programs, so viruses, instead of living in their own executable files, copy themselves into the middle of existing programs on the computer. This way, when we run our own program, the virus runs inside it.

This would leave the virus unable to run unless the right program happened to be run on the right day. To increase the chance of getting to run at the chosen time, the virus also self replicates into other executables on the same computer.




When you run a program containing a virus, it checks whether it is time turn run the payload yet. If so, it does. If not, it may search for an executable to copy itself into, to increase the chance that it will run at the chosen time, but then it will quickly return to the instructions of the program it is hiding in, so that you will not notice any difference, and the virus can continue to hide on your system.

Sometimes data files contain instructions that are run when the file is opened, to automate some part of dealing with the file – a macro. It would also work for a virus to locate itself in one of these macros. If the virus is on a network where it can get to executables on other drives on the network, it may get from one computer to another that way. If you use a removable drive with a program on it, the virus may copy itself into this, and then when you run the program from the drive on another computer, it may use that to copy itself to a program on that computer. Even if your removable drive has no programs on it, it contains a boot

sector, a small area where instructions to run automatically can be stored, and a virus might copy itself into this.

# Worms

Worms are another type of malware, different from viruses. A worm arrives on your computer in some way, usually via email or another type of messaging system, but sometime simply through being copied from computer to computer over a network. When the worm runs on your computer, it seeks other computers to send itself to. For example, if it arrived at your computer in an email, it may search your email address book for other people to email itself to. Then it sends copies of itself to multiple other victims, and continues the process on their computers.

Worms arriving by email could once count on being run automatically by the email program (an exploit). Most email programs now know better than to automatically open attachments, and so the email accompanying the worm must rely on some social engineering to encourage you to open it, giving the worm a chance to run.

A worm's whole payload is simply to self-replicate onto other computers. Once it has sent itself on, it has no reason to try to remain on your computer – it is not interested in destroying your data, just using your computer as a vector to reach others.

The goal of the worm is usually a denial of service, stopping access to the very service they use to replicate. Suppose that the worm is travelling by email. Email accounts live on a mail server, a computer whose only job is to deal with email. Email waits in your account until you download it. Every email, then, passes through at least one server (if both accounts are in the same domain, e.g. *owlmail.harford.edu*; it would go through two

servers if they are on different domains.)  As the worm bounces from one email account to another, each time it sends a copy of itself, that's another extra email the server has to deal with.  If a worm is making each account send an extra hundred emails, the server can become overwhelmed, having to deal with thousands more emails than it was designed for.  It may have to be shut down until all the extra email can be cleared out of it.

Even if there is no central server to take down, a worm copying itself from computer to computer over a network can overwhelm the network with too much traffic (copies of the worm) making communication between computers slower and slower until the computers can no longer communicate at all.

Denial of service is a temporary problem, but while the system is out of order, a business may lose money, health care data may be delayed, etc.  This denial of service may be just a matter of malice, e.g. a former employee with a grudge temporarily shutting down a business' email, but sometimes it is done for a profit; a business may be threatened with denial of service attacks unless it pays a protection fee.

## Adware

Adware makes money for its creator following the usual model for advertising.  A number of companies provide advertisements on the internet that they want people to see, and they pay a small amount every time someone is shown one of these ads.  We are used to viewing ads as part of viewing a video or visiting a web page – the content we are interested in is the lure to get us to view the ads, which help to pay for the distribution of the content.

Adware removes the content part, forcing the ads to pop up on a computer constantly no matter what else the computer is doing. The providers of the ads might not even know that the ad they provided was shown to you by malware instead of as part of a valid web page.

Adware has the strongest need for self-preservation techniques, because it cannot hide; the whole point of adware is that you *do* see it. Adware is not particularly interested in your data, it just wants to run on your computer.

# Spyware

Spyware's payload is to steal your information, to aid in the goal of stealing money. Spyware may send your files over the internet to its creator, and also do *keylogging*, recording all your keystrokes to send along as well. Based on this information, the spyware creator can find out your login usernames and passwords, and other personal information they can use[1].

Since spyware relies on you actively using your computer to get more information, it usually does a good job hiding. If spyware slows down your computer and you notice, you are less likely to continue typing in passwords and credit card numbers, so spyware that does so is badly designed spyware. Spyware certainly would not want to damage your data, since your data is exactly what it wants to steal.

---

[1] So you might want to be careful about any program installed on your computer (e.g. one you downloaded when you first used college wifi) that runs on your computer all the time (on campus and off) invisibly and can send any information back to its makers at any time and has also already been shown to have a major vulnerability to an exploit (https://www.eff.org/deeplinks/2011/10/safeconnect-universities-peer-peer-file-sharing)

# Bots

A malware bot is a program that runs on your computer to perform a task for someone else without your permission. In this case, the bot has no interest in stealing or destroying your data, it just wants to use your computer to do someone else's task.

Unless your computer is very fast, this may not seem very useful, but there are some tasks that are very well suited to parallel processing, so putting a thousand bots on slow computers all doing parts of a task at the same time may do a better job than one very fast computer. A group of bots working together on the same task is a *botnet*.

Sending spam is one example of a task that might be best done by a botnet, a denial of service is another. It also turns out that breaking the encryption used for internet commerce (e.g. to keep your credit card number secret) is another kind of calculation that could take months on a fast computer and hours via large botnet.

# Trojans

Trojan is sometimes used to refer to an approach to infection – if you install a program on your computer that has a virus already hiding in its executable, that may be called a trojan. However, Trojan also refers to a specific kind of malware that allows *backdoor access*, giving another computer the ability to take control of your computer remotely, looking at your data and using your system resources.

A Trojan that is running on your system could allow someone to remotely install another kind of malware on your system. It could also allow them to access your data and programs.

Primarily this is a problem for the high-profile and/or powerful: while being able to take control of the programs on a computer in charge of an electric power plant might be useful, taking charge of the programs on an individual person's home computer doesn't usually gain much that couldn't be gained by leaving spyware on their computer, and involves active work while the spyware would run on its own.

However, there is one resource that certain people want access to on many computers: the camera.  A trojan could allow a person to turn on your computer's camera (yes, possibly, without turning the activity light on) and watch you.

## Blended Threats

So far, we've talked about the characteristics of different types of malware.  Most real malware active now is a blended threat – it involves more than one type.  For instance, the payload of a virus could be to start a worm.  A worm could leave behind adware on each computer where it runs.  Spyware and trojans could be combined to steal information automatically but also provide the option to take over your computer actively by remote.