Interested in learning more about security?

SANS Institute InfoSec Reading Room

Virii Generators: Understanding the Threat

The most common generators are the virii script generators, polymorphic, and encryption generation engines. Each of these generators comes in multiple forms with multiple types of interfaces, the most common being GUI interfaces, command line interfaces, and assembly level interfaces. Thankfully, the government as has sought to deter the practice of virii creation through clear consequences spelled out for those who engage in such activities. Each of these precepts needs to be thought through more, however, to really u...

Virii Generators: Understanding the Threat

James Tarala
GSEC Practical Version 1.4
May 12, 2002

Abstract

Ever since Robert Morris unleashed his first Internet worm in 1988, virii have been a nuisance and a threat to both corporations and individuals alike. In the early days, worms such as these took an understanding of at least basic programming and of the vulnerabilities inherent in the operating systems at work in computer networks. Those virii that were released took time and effort to produce and often were not created with destructive or malicious intentions in mind. But that was then, and this is now. A lot has changed in the world in the past fourteen years. While in the past the novice would have no access to a common global network (the Internet), yet alone GUI tools to create, package, and distribute malicious code against any whimsical target, today even foreign pre-pubescents have the chance to annoy and harm the networked community at large.

For those security professionals charged with the task of protecting their corporate infrastructure, or for assisting the global community in defending against virii attacks, it is vital that there is a proper understanding of the threats which breed in the world. In the realm of virii, one of the vital understandings must concern virii generators and the risk that they pose to corporate and individual well beings. There are various aspects of this threat which must be understood in order to properly defend against it. First, one must understand where these kits are available, and how potential malware authors are able to obtain these generators. One must also understand the types of generators available, the simple script generators, polymorphic generators, and virii encryptors. Following in those steps, one must also see the ways in which these kits are utilized, through GUI menus, command line interfaces, and assembly generators. Finally, one must understand current trends in the world of virii authors, especially those contributing to their slowdown due to international legal ramifications.

It is only through thorough investigation and proper respect of an enemy that allows for an effective defense against such a foe. Thus it is only when the security community and system administrators at large understand the threat that virii generators bring to the table that they can begin to properly harden against any attack that they might bring. Applying software virii definitions, and proper engine updates is a crucial step in winning this war, yet it is only through proper understanding and education that it will finally be won.

Virii Generators: Understanding the Threat

James Tarala
April 23, 2002

On most any given day, America awakes to smell of fresh coffee, catches up on the morning's news, and prepares for a brand new day. Sadly, the brightness and hope a new day provides can often turn to fear and anxiety for the system administrator who awakes to the news of yet another computer virus discovered in the wilds of cyberspace. It turns out that yet another juvenile has been suspected of releasing code, composed by a kit downloaded from the internet that he has experimented with in his free time. And while the youth denies any malicious intent, the systems admin, newly out of bed begins to wonder… Is my network protected? Did I update my definitions? What about all of my clients? What began as such a peaceful day has now become a potential nightmare as the enterprise is scanned for the latest threat to the company's well being. Such begins the ritual too often begun in companies throughout America, often times due to virii created by individuals without any knowledge of programming, who have the desire to wreak havoc to the digital community around them.

If system administrators and the security community at large hopes to protect itself against such attacks, they need to understand the threat which confronts them. It is only through understanding that such a threat can be protected against. And while it is beyond the present scope to give a full and complete analysis of the issue of virii generators, there are certain basics which must be understood.

The most common generators are the virii script generators, polymorphic, and encryption generation engines. Each of these generators comes in multiple forms with multiple types of interfaces, the most common being GUI interfaces, command line interfaces, and assembly level interfaces. Thankfully, the government as has sought to deter the practice of virii creation through clear consequences spelled out for those who engage in such activities. Each of these precepts needs to be thought through more, however, to really understand the threat against the enterprise, caused by such virii generators.

Availability of Kits

Unfortunately while many users long for the illusion that such malware kits are confined to the dark, unreachable corners of the Internet, this belief is not grounded in reality. The truth is that any individual with access to a computer and the Internet can readily download dozens of generators within a short browsing session. While the popularity of such sites seems to be declining in recent months, the availability of these tools remains strong despite efforts directed against them.

There are multiple ways a would-be virii author could get their hands on such a product, none of which involves visiting a local Radio Shack™ (although sometimes it feels that easy). The four major ways to obtain such a package are:

- Through Internet web sites dedicated to virii which freely distribute the kits

- Through Internet web sites willing to distribute the kits, for a fee
- Through common IRC channels
- Through underground invitation IRC channels and ftp sites

Gone are the days when virii creation kits were freely distributed by web hosts willing to post any content for a fee. With increased federal regulation, harsher punishments for virii distributors, and ISPs willing to be more selective in what they post, fewer sites are conforming to the first two options listed above. While there are some groups still willing to post such materials for free, such as the Black Cat Virii Group ([BCVG] Network Security. 12 May 2002. http://www.ebcvg.com/viruses.php) and VX Heavens (VX Heavens. "VX Heavens: Binary." 12 May 2002. http://httpmirror.hwc.ru/vx.org.ua_80/bin.shtml), and more willing to sell their information at the right price, such as American Eagle Publications (American Eagle Publications, Inc. 12 May 2002. http://ameaglepubs.com/store/outlaws.html), their numbers are fading, leaving kit distributors to go more underground to peddle their wares. This leaves underground web and ftp sites and IRC channels as the most available channels remaining for anyone wishing to spread their code. While these mediums make it more difficult to locate the kits, it makes them available nonetheless.

One might be tempted as well to ask, why wouldn't the government simply crack down on those hosting or distributing these wares? Why wouldn't they make it illegal not only to propagate a virus, but also to distribute or host the creators of mass computer annoyance? Isn't that what the law calls being an accomplice? While these are all good and valid questions, there are no easy answers. Most would agree that virii code generators have very few legal or profitable values, legislators however often run into the roadblocks of First Amendment Rights and Internet activists attempting to preserve the 'freedom and purity' of the net whenever attempting to ban any such site online. As network administrators and citizens of an interconnected world, the world can only hope however, that lawmakers will one day soon take up the battle against such proliferators of digitalized harm and at least slow the propagation of these virii in the wild. Thankfully many web sites and print media organizations, including semi-militant publications such as Soldier of Fortune magazine, have banded together to ban the advertising or sale of the for-profit compilations of computer virii, virii creation labs, and manuals dedicated to their creation (American Eagle Publications, Inc. 12 May 2002. http://ameaglepubs.com/store/outlaws.html). While most recognize that such actions will never eliminate the spread of such kits, anything which can slow down their reproduction is appreciated.

Types of Kits Available

As one examines the dozens of virii creation kits available online today, one notices three major categories of creation kits available for download. The first is the standard script generating kit that can be used by non-programmer types to release virulent code into the wild, such as worms, Trojans, and virii. This type, as will be examined, tends to be the tamer of the two virus generators, and tends to be used primarily as a shortcut to quickly produce the desired code. The second type is what are called polymorphic generators. These generators produce code which is similar to the standard script generators, acting as a shortcut to producing malware code which can be released to the world. The difference with this type of kit is that the virii created are

intended to mutate themselves and change their behavior at each infection along the way, thus being more difficult to detect their signatures. The third type are virii encryptors. These final kits are used to do just as their name suggests, encrypt the scripts already generated by another source in an attempt to change their properties and hide commonly detected signatures for software developed to detect them.

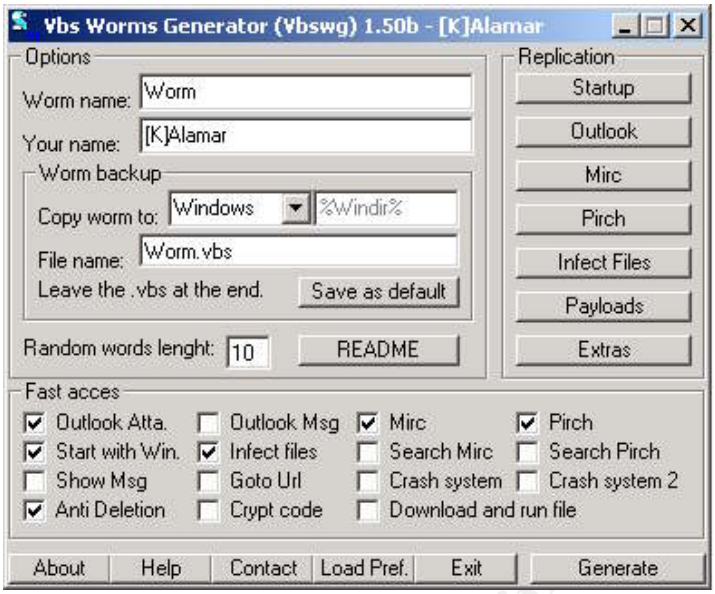
Simple Virii Script Generators

The majority of the virii generators available for use today fall into this first category. Virii script generators are primarily used by non-programmers who are looking to experiment in the world of virii creation, without a knowledge of programming techniques. Rather than learning the programming languages themselves, users of this type of kit will often simply follow the well documented instructions for a given tool to begin the process of creating a product of mass annoyance in point and click fashion. Gone are the days when virii authorship mandated the knowledge of assembly, C, or Visual Basic Script. Now, all that a virus creator must do is simply know where to obtain such a tool, install the software, and follow the appropriate selections to have a virus of his very own.

The majority of the individuals utilizing these tools are relatively young, inexperienced, or uninitiated into the world of viral destruction. This is due to the fact that these simpler kits are often easier for virus scanners to detect than the polymorphic or encrypted virii generated by other kits. And even though most serious virus authors would hurl angry remarks and classifications at these 'students' of code, they often can do enough damage to sufficiently annoy end users and cost corporations millions in often unintended damages.

The prime example of a virus created in such a fashion is the now infamous Anna Kournikova worm. This virus effectively spread through millions of computers worldwide, avoiding even the best anti-virus tools through the simple use of a GUI creation tool. The tool used with this particular virus was the VBS Worm Generator, created by [K]alamar, believed to be a seventeen year old resident of Buenos Aires. A twenty year old Dutch resident, calling himself 'OnTheFly' used this tool to exploit users with a relatively harmless payload, thankfully only directing users to a Dutch computer shop's web site in January of the following year (Grazi, Alberto. "VBS Worms Generator." 21 Feb 2001. 12 May 2002. http://rr.sans.org/malicious/VBS_worms.php). The accused 'OnTheFly' has since been found and been taken into custody by local law enforcement, yet the damage has been done.

The following is a screen shot taken from the VBS Worm Generator, illustrating the relative ease by which an individual could create a virus. More will be discussed about this simple process later.



(Screen shot taken from actual VBS Worms Generator application, version 1.50b. Application was downloaded from VX Heavens. "VX Heavens: Binary." 12 May 2002. http://httpmirror.hwc.ru/vx.org.ua_80/bin.shtml)

Thankfully [K]alamar recognized the potential havoc which could be reeked through the use of his application and at the urging of friends after the initial outbreak of the Anna virus, [K]alamar took down the site where he was hosting the VBS Worms Generator ([K]alamar's web site, http://virii/at/k/ , was still down as of 12 May 2002.), although development of the tool continues and can be found with a newer version on many web sites today, with the code base still being maintained by [K]alamar. Unfortunately for the world, the damage had been done, and the kit spread through multiple channels throughout the web, and is still one of the more popular choices for experimentation with worm generations.

Some other examples of script generating virus kits are as follows:

- Virus Creation 2000 System
- Virus Generator
- Satanic Brain Virus Tools
- WinScript Virus Kit
- Ye Olde Funky Virus Generator
- Access Macro Generator
- Worm IRC Script Kit
- VBC Worm Coder
- Goofy Batch Virus Generator
- BioHazard Worm Generator

While this list certainly is not complete, considering those such as Jack Clark, the European product manager for Network Associates, believes that there are at least 100 various virus generators being circulated today. Unfortunately this number continues to grow in leaps and bounds each year (Leyden, John. "Virus toolkits are s'kiddie menace" 21 Feb 2001. 12 May 2002. http://www.theregister.co.uk/content/archive/17106.html).

Polymorphic Virii Generators

The second type of virus generators circulating around the Internet today are the polymorphic virus generators. The purpose of these generators is simple, while the majority of virii created today are easily detectable by anti-virus tools, due to their predictability and similarity in structure, polymorphed virii are purposely engineered to avoid detection by anti-virus systems anxiously awaiting their arrival. The virii themselves are designed with this purpose in mind and have best been defined by Sha Sha Chu, who states that polymorphic virii are "viruses which change slightly each time they are executed. These are meant to defeat anti-virus scanners which search for certain strings of code to identify viruses (Chu, Sha Sha. "Virus: A Retrospective." 12 May 2002. http://cse.stanford.edu/class/cs201/projects-00-01/viruses/viruses101.html)."

The attempted stealth used by these polymorphed virii is not the major cause of alarm, however. The goal of most virus authors is to propagate their code as extensively as possible, and to obtain the supposed fame which is to follow, often by executing a rather tame set of instructions to be executed on a computer. However, those who enter into the arena of polymorphed virii often have more destructive purposes in mind. Not only are these authors attempting to slip by the virus intrusion systems undetected, they often are going through this effort in order to deposit a payload which are more destructive by nature. Low level drive formatting, MBR destruction, system file deletion, and data destruction are common goals of this type of virulent.

While the goal of the virus script generators is simply to produce the code which could infect a system, the polymorphic generators attempt to write code which will not only deposit the intended destructive payload, but will deceive the operating system and anti-virus software into thinking that the instructions being executed are good and normal commands to be run on a given machine. The difficult aspect of these virii has been that they are often successful in their desired purpose and the danger remains that the more complex the polymorphs become, the more susceptible systems will be to attack.

Examples of polymorphic generators are fewer, yet a few of those available are:

- Trident Polymorphic Engine
- Dark Avenger's Engine
- VICE V0.2A
- Duke's Polymorphic Generator

Virii Encryption Generators

The third and final type of virii generators are the virii encryptors. These generators are used in conjunction with other virii, worms, or Trojans which have been created in order to hide the signature from virii detection software, which is often the end goal of virii creators. These generators will run 'on top of' the code already written by an individual (by hand or through another generator), and attempt to change the code's properties enough, and encrypt the commands enough to mask the activities being attempted by the code. Often authors have tried taking already used virii code, and run it through one of these types of generators in an attempt to increase the infection from a particular virii, or in essence re-use the code to their advantage. The encryption used with the virii often masks the code from upper level scanners, and requires anti-virus software to detect the code on a per workstation basis. Fortunately for the diligent administrator, most anti-virus applications are intelligent enough to detect when malicious code is attempting to execute on a machine, and thus protect that machine from infection.

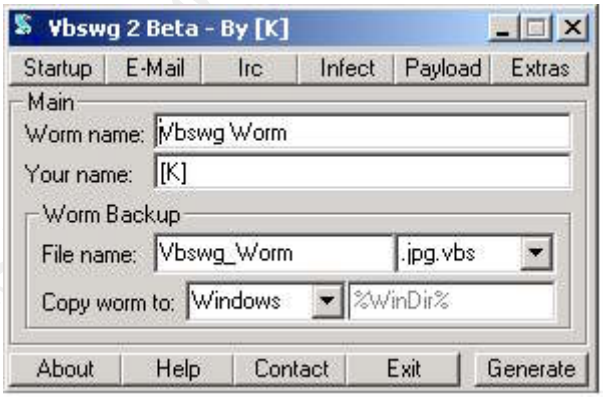
Brief Tutorial on How to Use

While finding, classifying, and fearing these virii is one leg of the journey towards understanding, actually seeing them in use is vital when obtaining a knowledge of their potential danger. In the present scope it is impossible to walk step by step through the process of creating a virus with many of these tools, however it is viable to examine the types of interfaces available and to understand the general concept of what it would take for someone to create a virus. In embarking on this quest one will typically see three types of generator user interfaces available, GUI creators, command line creators, and assembly level creators, each of which must be examined individually to understand its use.

GUI Generators

The most visible generators available to the world today fall into this first category of creation kits. This is the generator type distained and reviled by major newspapers and media outlets as scourges of the modern world. This is also the type of generator which is the most readily utilized by the hacker newbie who is still wet behind the ears in his potentially destructive habits. Two examples of this type of generator, which will be examined in detail below, are the VBS Worm Generator and the ANSI Bomb.

The first to be examined is the VBS Worm Generator, written and maintained by [K]alamar, which can be credited for the 'successful' Anna Kournikova virus discussed earlier. While documenting a tutorial on the document is beyond the present scope, a basic understanding can be obtained through an examination of the current version's main dialog box as seen below.

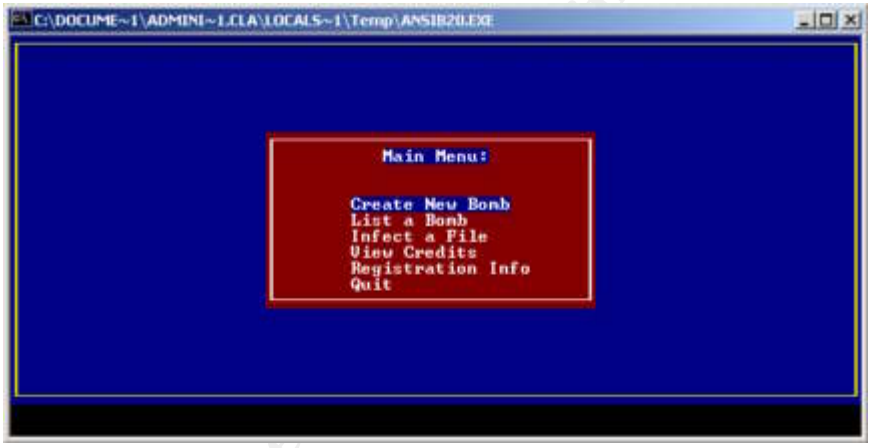


(Screen shot taken from actual VBS Worms Generator application, version 2 beta. Application was downloaded from VX Heavens. "VX Heavens: Binary." 12 May 2002. http://httpmirror.hwc.ru/vx.org.ua_80/bin.shtml)

As can be seen from the screen shot, this generator is a windows based generator which allows the user to simply make selections in a point and click fashion, and then compile (Generate) the virus for distribution. There are three major modes for the virus' distribution, Startup, E-mail, or IRC, which can all be used in tandem, as noted on the top menu bar. When the user clicks on any of these methods a dialog box appears, allowing the user to customize particular choices to his

specifications. The other major options the interface allows for is the payload to be delivered, virus name, file name, file extension, and OS to infect. Virus creation doesn't get any easier than this. Thankfully for white hats, however, just like those who create the virii have access to this software, so do the research teams at the major virus detection research labs, making a virus which has been generated by this tool easy to spot and identify in the wild. The general rule is that the easier to use and more wide spread the virus generator is, the more likely that the effects to diligent individuals using virus detection software will be minimal.

A second type of GUI user interface available with some generators is illustrated below by the ANSI Bomb 2000 (ANSIB20.exe). This is a typical text based menu utilized by many text only operating systems and software applications today. While not as familiar to many of the younger would be attackers, to most utilization is simple. To create a bomb, edit a bomb, or infect a file all someone must do is use their keyboard to negotiate around a screen selecting choices and inputting parameters to generate the desired results. The use of such a tool is straightforward to any desirous of making such a tool.



(Screen shot taken from actual ANSI Bomb 2000 application. Application was downloaded from VX Heavens. "VX Heavens: Binary." 12 May 2002. http://httpmirror.hwc.ru/vx.org.ua_80/bin.shtml)

Command Line Generators

The second type of generator available today is the traditional command line tool which enables users to generate a virus using a combination of command line tools and switches to generate their desired results. These command line tools work in a similar manner as any other command line tool would in Unix, Windows, or Dos. The general operation of these commands involves issuing a command, and then following that command with switches which pass variables to the command and allow for customization of the outputted results. An example of such a command is as follows:

```
ls -al
```

This standard Unix statement issues the command, `ls`, which lists the files and folders in a given directory. The following component, `-al`, indicates a switch through the use of the character '`-`' or '`/`' followed by the particular switch recognized by the command, in this case '`a`' and '`l`', which lists all the contents of the current directory and lists them in long format, respectively. Thus the outputted results of such a command would be to list all the contents of the current directory in long format.

Command line virus generators function in this same manner, allowing the user to simple pass arguments to the command using the switches and thus customizing the code which comes as a result. Again due to the static nature of such commands, many virus research labs are able to detect the signatures of the virii created. However, the more customizable and obscure the generator is, the more likely it is that a virus created by the tool will slip under the radar of a detection system. For these reasons often the command line generators produce code which is more dangerous to the general public who is attempting to be diligent with the protection of their systems.

Assembly Level Generators

The final major type of interface available for the generation of virii is the assembly level generators. These generators are by the far the most difficult of the three types to use, and actually require the user to have a general knowledge of assembly level programming before the interface can be used. Such software packages must be installed and utilized separately from the generator, and while more difficult to use, their payloads can also potentially cause an increasing amount more damage to its prey.

Unfortunately, while these virii are more difficult to create, there are enough entry level tutorials available on the web to teach newbies how to form these dangerous creations, without even a full understanding of what could happen as a result. Many are the newbie computer hard drives that have been destroyed through their own ignorance. The down side to this level of generator is that once these virii reach the wild, they often can inflict the same payload on their unsuspecting host, potentially causing damages from mere annoyances to actually physically disabling computer devices. Either way the thought can be frightening.

Virus Creation Kits… the Legal Side

Realizing the potential impact of all that has been discussed previously, what stops the world's information highways from becoming overloaded with viral vehicles? How is it that mail servers are even able to function day to day without becoming overwhelmed by the onslaught of malware attacks? If creating a virus really is this easy, why isn't everyone doing it? The answer to this question for now would appear to be fairly simple. The reason why more virii aren't found assaulting our digital landscapes is due to the legal and criminal ramifications of releasing such code into the wild. With the arrests of the authors of highly visible virus attacks such as the Anna Kournikova worm, The Love Bug, and Melissa making the nightly news, would be attackers are becoming more and more hesitant to release their creations into the world. Following a principle understood since ancient times, if a government wants to stop a particular behavior from

occurring in their realm, simply create and follow through on a law which makes the undesired behavior simply not worth the consequences attached to the action.

Recent events should help us to see some of the legal ramifications for some of these virii which have reached the real world. According to the Computer Crime and Intellectual Property Section (CCIPS) of the U.S. Department of Justice, for "whoever willfully or maliciously injures or destroys or attempts willfully or maliciously to injure or destroy any of the works, property, or material of any radio, telegraph, telephone or cable, line, station, or system, or other means of communication…or willfully or maliciously interferes in any way with the working or use of any such line, or system, or willfully or maliciously obstructs, hinders, or delays the transmission of any communication over any such line, or system (Department of Justice: Computer Crime and Intellectual Property Section. "18 U.S.C. 1362: Communication Lines, Stations, or Systems." 24 Apr 2000. 12 May 2002. http://www.cybercrime.gov/usc1362.htm)," (think spreads a virus) is eligible for up to ten years in a federal penitentiary and/or fines appropriate to the damages caused by such a violation. And if anyone thinks that the federal government will not do their part to put a stop to such computer crimes, simply ask Herbert Pierre-Louis (directed virus against Purity Wholesale Grocers), Reomel Ramones (The Love Bug virus), or David Smith (Melissa virus) whether or not they think the U.S. government is serious about this issue (although one would need to ask them in their respective prisons). And as if this were not enough to cause a would be attacker to seriously question whether or not he would be willing to release a virus into the wild, the recently passed Patriot Act now leaves the door open for such activities to be labeled as an official act of terrorism against the people and government of the United States (with full retribution and punishment to follow).

Of course many may argue that this is a simplistic explanation for the relatively low number of virii in the wild (compared of course to the potential). Many programmers would state that this fact is really due to a collective community conscience. They would argue that virii creators really have no intention of hurting anyone, and are typically morally upright citizens just trying to live in peace (note typically). Such proponents would state that the whole purpose for virii in the first place is simply for learning purposes. These generators, proponents would say, enable relatively inexperienced programmers a faster learning curve into the world of creative application development. These same individuals will point to what has been considered by many to be the author of the first Internet worm, Robert Morris, who in 1988 as a doctoral student at Cornell University released the first Internet worm into the wilds of the then young Internet, infecting a mere 6,200 systems, yet causing upwards of $15.5 million (Dewing, Scott. "Virus Writers: Who Writes This Stuff Anyways?" 2002. 12 May 2002. http://www.projecta.com/Page.asp?NavID=250). The question to those advocating this philosophy should be, what ever happened to "Hello World"? Why has it now become necessary to examine malicious code, rather than ADO or SQL statements to have a better insight into the world of application development? Thankfully most professionals are able to see through the smokescreen generated by this camp and recognize the potential destructiveness and danger in such an approach. Simply put, "When playing with fire, you're going to get burned." Even Robert Morris found this out personally when it was discovered that the code which was originally intended to simply prove a point, brought down corporations and portions of the government to satisfy intellectual curiosity.

This is not to say that many programmers do not have a conscience regarding such activities. In fact, overall, the majority of coders today recognize the potential danger in creating and releasing, even accidentally, a virus into the world. Not only are they potentially causing damage to national infrastructure, corporate data and resources, and personal computers worldwide, but they personally are performing an act which will bring out much more than a simple laugh and slap on the wrist. And while most programmers will admit to at one point in their lives creating a virus, simple as it may be, only the truly destructive are willing to allow their creations into the world.

Conclusion

There can be no question that ever since Robert Morris released the first Internet worm in 1988 there has been a rapid increase in the proliferation of computer virii in the world. These virii range from annoying pests to full blown destructive forces and yet as a whole have caused in the millions of dollars in damages in the past 14 years. As the spread of these virulents continues, it is being recognized more and more that those virii which have made it loose into the wild are not simply the work of destructive malcontents, but also of what society would consider 'normal' individuals experimenting with the use of virii generators easily found online.

As has been noted already, virii can be found in a multitude of locations online and are readily available to even the novice programmer. There are many types of virii kits available, but the most common are the virii script generators and polymorphic generation engines. Each of these generators utilize multiple types of interfaces, the most common being GUI interfaces, command line interfaces, and assembly level interfaces. Thankfully, as with all virii released into the world, the legal environment is becoming more and more strict with the release of new strains of code into the world. Thus the public has been protected by the government as they have sought to deter the practice of virii creation through clear consequences spelled out for those who engage in such activities.

As long as people are relying on computers for business or personal use, there will always be people who attempt to use their knowledge (or other people's knowledge) of the interworkings of these machines against others. And while there is no way to utterly stop the spread of malware in the world, there are many steps one could take to be diligent to protect oneself. There is no safety in obscurity as many wish was true, as evidenced by the attacks of 2001 (Code Red, Nimda, Code Red II). As system administrators and as a part of the larger digital community everyone needs to do their part to at least slow the spread of these dangerous infectants.

Sources Referenced:

American Eagle Publications, Inc. 12 May 2002. http://ameaglepubs.com/store/outlaws.html

[BCVG] Network Security. 12 May 2002. http://www.ebcvg.com/viruses.php

Chu, Sha Sha. "Virus: A Retrospective." 12 May 2002. http://cse.stanford.edu/class/cs201/projects-00-01/viruses/viruses101.html

Delio, Michael. “You, Too, Can Write an Anna Worm (Part 1).” 15 Feb 2001. 12 May 2002. http://www.wired.com/news/technology/0,1282,41817,00.html

Delio, Michael. “You, Too, Can Write an Anna Worm (Part 2).” 15 Feb 2001. 12 May 2002. http://www.wired.com/news/technology/0,1282,41817-2,00.html

Department of Justice: Computer Crime and Intellectual Property Section. “18 U.S.C. 1362: Communication Lines, Stations, or Systems.” 24 Apr 2000. 12 May 2002. http://www.cybercrime.gov/usc1362.htm

Dewing, Scott. “Virus Writers: Who Writes This Stuff Anyways?” 2002. 12 May 2002. http://www.projecta.com/Page.asp?NavID=250

Fansler, Bryan. “Virus Generators and Their Implications.” 19 Feb 2001. 12 May 2002. http://rr.sans.org/malicious/generators.php

Grazi, Alberto. “VBS Worms Generator.” 21 Feb 2001. 12 May 2002. http://rr.sans.org/malicious/VBS_worms.php

Leyden, John. “Virus toolkits are s’kiddie menace” 21 Feb 2001. 12 May 2002. http://www.theregister.co.uk/content/archive/17106.html

Pearson, David. “Psst…Hey Buddy, Wanna Create a Virus?” 5 Dec 2001. 12 May 2002. http://rr.sans.org/malicious/create.php

Salo, Markus. “Dark Side of the Moon: What Motivates Virus Writers.” 1994. 12 May 2002. http://vx.netlux.org/texts/html/darkmoon.html

VX Heavens. “VX Heavens: Binary.” 12 May 2002. http://httpmirror.hwc.ru/vx.org.ua_80/bin.shtml

VX Heavens. “VX Heavens: Library.” 12 May 2002. http://vx.netlux.org/lib_diff.shtml

Last Updated: November 20th, 2017

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS San Francisco Winter 2017	San Francisco, CAUS	Nov 27, 2017 - Dec 02, 2017	Live Event
SIEM & Tactical Analytics Summit & Training	Scottsdale, AZUS	Nov 28, 2017 - Dec 05, 2017	Live Event
SANS Khobar 2017	Khobar, SA	Dec 02, 2017 - Dec 07, 2017	Live Event
SANS Munich December 2017	Munich, DE	Dec 04, 2017 - Dec 09, 2017	Live Event
European Security Awareness Summit & Training 2017	London, GB	Dec 04, 2017 - Dec 07, 2017	Live Event
SANS Austin Winter 2017	Austin, TXUS	Dec 04, 2017 - Dec 09, 2017	Live Event
SANS Frankfurt 2017	Frankfurt, DE	Dec 11, 2017 - Dec 16, 2017	Live Event
SANS Bangalore 2017	Bangalore, IN	Dec 11, 2017 - Dec 16, 2017	Live Event
SANS Cyber Defense Initiative 2017	Washington, DCUS	Dec 12, 2017 - Dec 19, 2017	Live Event
SANS Security East 2018	New Orleans, LAUS	Jan 08, 2018 - Jan 13, 2018	Live Event
SANS SEC460: Enterprise Threat Beta	San Diego, CAUS	Jan 08, 2018 - Jan 13, 2018	Live Event
SEC599: Defeat Advanced Adversaries	San Francisco, CAUS	Jan 15, 2018 - Jan 20, 2018	Live Event
SANS Amsterdam January 2018	Amsterdam, NL	Jan 15, 2018 - Jan 20, 2018	Live Event
Northern VA Winter - Reston 2018	Reston, VAUS	Jan 15, 2018 - Jan 20, 2018	Live Event
SANS Dubai 2018	Dubai, AE	Jan 27, 2018 - Feb 01, 2018	Live Event
SANS Las Vegas 2018	Las Vegas, NVUS	Jan 28, 2018 - Feb 02, 2018	Live Event
Cyber Threat Intelligence Summit & Training 2018	Bethesda, MDUS	Jan 29, 2018 - Feb 05, 2018	Live Event
SANS Miami 2018	Miami, FLUS	Jan 29, 2018 - Feb 03, 2018	Live Event
SANS London February 2018	London, GB	Feb 05, 2018 - Feb 10, 2018	Live Event
SANS Scottsdale 2018	Scottsdale, AZUS	Feb 05, 2018 - Feb 10, 2018	Live Event
SANS Secure India 2018	Bangalore, IN	Feb 12, 2018 - Feb 17, 2018	Live Event
SANS Southern California- Anaheim 2018	Anaheim, CAUS	Feb 12, 2018 - Feb 17, 2018	Live Event
SANS Dallas 2018	Dallas, TXUS	Feb 19, 2018 - Feb 24, 2018	Live Event
SANS Secure Japan 2018	Tokyo, JP	Feb 19, 2018 - Mar 03, 2018	Live Event
Cloud Security Summit & Training 2018	San Diego, CAUS	Feb 19, 2018 - Feb 26, 2018	Live Event
SANS Brussels February 2018	Brussels, BE	Feb 19, 2018 - Feb 24, 2018	Live Event
SANS London November 2017	OnlineGB	Nov 27, 2017 - Dec 02, 2017	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced