Global Information Assurance Certification Paper

The Davinia virus - can we defend against the latest attack vectors?

So there's another new virus out, so what? I'll download the virus update file, make sure it gets propagated out to the workstations and my e-mail server and worry about the 50 other things on my to-do list for this morning. For anyone who has network administrator as part of his or her job description, I'm sure this sounds all too familiar. Viruses have become a commonplace fact of life for anyone who uses a computer and little more than an item on an update checklist for administrators. Fortunately for most administrators, the user population has become educated enough to realize that just randomly opening attachments is a bad thing. We have Melissa and the I Love You virus to thank for that, and it was a lesson from the school of hard knocks. Unfortunately we may have another lesson coming our way in the form of the children of the Davinia virus.

Viruses in the past have been pretty straightforward - someone brings a floppy along or sends an e-mail that has a file that's infected. The problem with a new virus isn't how to detect it, but merely a question of how fast that the signature can be distributed to the user community. There have been many variations of this simple infection method, like scripts than run as soon as an E-mail is opened and macros that hide inside files that you would expect to get. Lately there have been a rash of files with double extensions to try to fool people into thinking the file is something they would want to see or is not a file type that would not normally carry a virus, like the AnnaKournikova.jpg.vbs seen very recently. As with most other viruses, a signature has been distributed to take care of this threat and it has been reduced.

Unlike most of the viruses that we have seen to date, however, the Davinia does not use an attachment at any point of its lifecycle. Technically, Davinia is a worm, since it is self-replicating. The method that this worm uses is very elaborate, and as such avoids almost all types of detection.

It should be noted that as of the writing of this paper, the Davinia virus has been rendered inert. It represents no further risk in its original form. It's methodologies are described here as a lead into the discussion that follows in an effort to help defend against the imitations that are sure to come.

As with most viruses nowadays, it exploits a vulnerability in Microsoft Outlook. Microsoft has acknowledged that there is a problem, and has posted a fix on it's website. The list of programs that are actually affected by this vulnerability is: Access, Excel, FrontPage, Outlook, PowerPoint, Project, Publisher, Word, the Works Suite and PhotoDraw. They are calling it the 'Office 2000 UA Control Vulnerability'. According to Microsoft, there is an ActiveX control, Ouactrl.ocx, which was included with Office 2000 that is incorrectly marked as "safe for execution". The control, known as the Office 2000 UA Control, allows for the scripting of Office 2000 functions, used in the Help and Show Me functions. It is important to know that this control is not included in SR-1 of the Office 2000 package, which means that SR-1 is not susceptible to this type of attack. The easiest way to verify if your machine will be susceptible is to check the version of the Ouactrl.ocx. If you have version 1.01.0009 or 1.0.1.9, then you are vulnerable. Version 2.0 is the one that is needed to protect your machine. If the office install is in the default directory, then the ocx will be in C:\Program Files\Microsoft Office\Office.

Now that we know what allows the attack, lets take a look at how the worm works. There are actually two stages to the delivery of this worm. The first stage of the infection is an E-mail containing HTML script. The E-mail itself seems to be blank with even the subject line being empty. However, when it is opened the HTML script takes advantage of the UA Control Vulnerability to launch Internet Explorer. This is the second phase of the delivery. The newly spawned Internet Explorer attempts to contact a web site to download a Word 2000 document named LD.DOC that contains a macro. The first thing that the macro does is to disable the Word Macros protection to allow the rest of the macros to run undetected. This is also accomplished via the UA Control Vulnerability. With this accomplished, the Word Macro links into the Outlook address book and E-mails itself out to everyone in there. The meat of this virus however is the script that gets created on the hard drive. LITTLEDAVINIA.VBS is written after the e-mails are on their way. The last thing the macro does is to change the registry so that the vbs script will run the next time the PC is rebooted. It makes an attempt at limiting the possibility of discovery by creating a registry key indicating that it is an infected machine. If this key exists, no further e-mails will be sent, regardless of the number of times DL.DOC is run.

The littledavinia.vbs script, when run at the next reboot will potentially do two things. It may, depending on the variant, display an HTML document containing a customized message to the user, in Spanish. The second thing, which always occurs, is the overwriting of all the files on local and mapped drives with the same message. The files will also be renamed. If left on it's own, this process will destroy all the files on the local and network drives.

The ease with which this virus was disabled may give the computer community a false sense of security about this new hybrid transmission method. All that is needed is to find the web site and shut it down. Of course, that is assuming that there are no more advances made in this vein of attacks, which I personally find very hard to believe. Considering the almost weekly developments of new Distributed Denial of Service attacks, it is not too far a stretch to believe that some kind of zombie community could be created to operate as download sites for the infected Word doc.

Given the delivery method, it will be remarkably difficult to block the distribution of the E-mail. So what do we do? Since the virus attacks using Microsoft Outlook, there is no way to use an application proxy to block any of the delivery methods, since the virus uses standard

ports and applications to propagate. Virus scanning technology does not currently allow for URL link blocking, although this will most likely be the developed method of defending against these types of attacks. There are already web-blocking programs available on the market, such as SurfControl and WebSense that do allow for URL blocking, and these might be the best defense mechanism against this type of virus. The problem is that most of these programs would have to be updated on a per incident basis by local administrators. These web-blocking programs work in one of two ways. The first method is content filtering where pages are scanned for certain key words or word combinations. The second, method, which is becoming more popular, is actual URL verification. This is accomplished by the use of a database of URLs that are divided up into categories by the software vendor. The administrator then picks and chooses the categories that are to be blocked. Nightly updates allow for new URLs to be categorized without the administrator's intervention. This second method would be most effective against this new form of distribution, but it would actually require the administrator to set up the URLs in a custom category on a case-by-case basis, because there is currently no predefined category for viruses or hostile web sites. If the software vendors could be convinced to add this type of category then the burden of updating would be removed from the administrators shoulders, making this type of defense much more effective. Of course virus-scanning software blocking URLs and web-blocking software checking for viruses might cause a few anti-trust remarks, but when we are dealing with a cross-category problem, we need a cross-category solution. Having both software packages blocking the URL's would be the ideal defense here.

With everything that has been discussed so far, however, we are still only talking about a *reactive* solution. If we are ever to get ahead of these new advances in virus development, we will need to start planning a *proactive* defense.

There is an argument against the amount of time that is required to set up such a defense. Is the time spent in man hours setting up an elaborate set of defenses really worth what it protects? Like everything else in network security, the answer is: it depends. If the data being protected is of a fairly static nature and having a tape backup is an acceptable form of data recovery, then no, the time spent hardening the defenses around it is probably not justified. However, if the data is the work of thirty-five developers, all stored on the same network drive and the loss of that data would result in the loss of possibly 280 man-hours (8 hours x 35 employees), then maybe it is worth taking a little extra time to protect the information.

So what other tools are out there that can help when the best that virus software can offer is a new signature file *n* hours after the first reported infection? The answer may surprise you.

Since Davinia - along with so many other popular viruses - is an Office virus, lets take a look at what Windows products and procedures are out there to help us out. Unix users can relax and dream of the days when they can actually choose not to use Office for UNIX 2002.

The very first product that can help is Outlook. Yes, the very culprit that causes most of the trouble can also prevent a lot of it. In Outlook 98, there was a flag that you could check off that prevented delivery of outgoing messages until the Send/Receive command was issued. Until the user actually clicked Send/Receive, any outgoing messages were stored in the outbox. With Outlook 2000 that option has been taken away, but there is a very simple Rule that can be set up to mimic this functionality. Microsoft has an article on their website describing exactly how to set it up. This allows users to see when unexpected E-mails are being cued up. It also gives them the chance to delete the E-mail before it can be sent out thus preventing propagation all together. This does not prevent the user from infecting his or her own machine, but it does put a serious damper on the lifetime of the virus and that is a good place to start.

So now lets look at protecting the user. How many network administrators set up users as being administrators of their local machine? Sound ridiculous? Ever had to support a user community made up of 80% development staff? The only way to stay sane is to give them at least Power User privileges, which on a local machine may as well be administrator. And how many admins set their own accounts up as Domain Admins? It has to be that way, doesn't it? A user calls up because their account is locked out, so you need to have Domain Admin rights to reset them. With UNIX, this is easy because you can quickly use the SU command to have the administrator priveledges needed to complete the fix. With Windows, there is no SU. You have to log off and log back on, which, let's face it, is not the fastest process in the world. Enter Virtual OS software. VM Ware is a good example of this type of application. It is possible, using this software, to have two or even three virtual machines running on the same PC. So you can log on using a regular user account, with limited domain access, for your E-mail and Word use and then have a separate virtual machine logged on with an Admin account for the system support. The key is that you ONLY run system utilities, such as USRMGR or SVRMGR under this account. So if something bad does come into your e-mail, it doesn't get as far as if your were logged on as the administrator. This is useful for the admins in the company, but may not be financially feasible for the whole company.

Lets take a look at why someone might need to have admin rights on their machine. For a developer, it is usually because they need to tweak the settings of their development apps, such as SQL Server or IIS. They usually end up needing to create registry settings as well. For an average user working in the accounting department, there is no reason at all for them to have admin rights. Almost any application updating that needs to be done would be on a server anyway. So lets take a look at development for a minute. If it's not cost effective for each developer to have virtual machine software on their desktop, then the same is probably true for apps such as SQL Server. Having a development server that houses just these applications - no source code - would be a likely solution. All the developers can be given full rights to the server, so they can make any changes necessary and still have their desktops secured. The registry entries are a little trickier, but can still be handled. It is possible for an Admin to create a Key within HKEY_LOCAL_MACHINE/Software - call it Dev - and give Domain Users full control of it. Sysdiff, a program found in the NT Server resource kit for recording changes made to any NT machine, can actually be used to propagate this change to every development machine. The end result of this is that all developers have a Registry Key that they can play with while leaving the rest of the registry secure. Obviously there may be a little bit more involved in getting the developers to go along with this, but explaining the loss of productivity and profit that could result from failing to secure the desktops may

help sway upper management to back your plans.

Sysdiff is actually a very powerful tool in application rollout schemes. If you have an application, say IE5, that you have set a fair number of custom settings on, like the Internet Security level, and you want everyone to have these settings, then creating a Sydiff file that contains the whole install of the application can drastically reduce rollout time.

Hard drive cloning can also be a fast way of distributing base systems already preconfigured to secure specs. Programs such as Norton Ghost are excellent at this. It is a much faster process than the Sysdiff method, but it is also very hardware specific, whereas Sysdiff is hardware independent.

The last distribution tool I will discuss is Microsoft's System Management Server 2.0. SMS is a tool that allows for incredible control over what goes on at a user's desktop. It allows for automated software distribution, useful for applying the latest Microsoft security patch, as well as software monitoring. It would be possible using SMS to find out exactly how many machines actually had the DL.DOC file without having to leave the console machine. An admin can track hardware as well as software and even take remote control of a users desktop, right from their terminal. It is possible to accomplish all of this with other commercial product, such as Symantec's PC Anywhere (www.symantec.com) for remote control and WinWhatWhere Corp's WinWhatWhere Investigator (www.winwhatwhere.com) for software monitoring, but all of this functionality is in one place with SMS.

Of course, with something this powerful, there has to be a drawback back, right. Actually yes there is. There is a fair amount of preparation work involved in setting up even a basic SMS server. It is one of those Microsoft products that takes a lot of time to become specialized in. It is also one that can lend itself to further security issues. Having this amount of control obviously requires Domain Admin rights, which the SMS user accounts have. And being a Microsoft product, we know that there will be bugs in it. This is not criticism; it is a fact of life. However, with the proper amount of preparation, both in experimentation and planning, an SMS server can become a valuable ally in the battle against security attacks.

Since we discussed source code as being something of great value to a company, lets take a look at how to secure it. It is not possible to prevent developers from updating source code. It's their job. So given that we have to trust our co-workers at least some of the time, we can still protect a copy of the file. Visual Source Safe is another Microsoft product that allows source code to be checked-out. Basically the code is stored in a secure database until someone needs to make a modification. The checked out code is really a copy of what is in the database for the user to play with. Once the changes are considered finalized, the user must check the code back in. So if something were to happen to the checked-out code, the repository is still secure on a network server that the developers only have access to thru Source Safe. Of course, there are other programs on the market that mimic this, but Microsoft's version integrates seamlessly with Visual Studio.

So, after all of these security updates, can we start to feel secure? Well, no, but at least we can go home at the end of the day with the knowledge that we have done our part in demonstrating Due Care. It is inevitable that an attack will eventually penetrate even the best of defenses, but we don't have to sit back and throw up our hands in defeat. If we can find new methods to protect systems before they are compromised - and this may even be using existing defenses in new ways - then we may actually be able to stay ahead of the black hats. If we can enlist the help of software vendors to implement new functionality, such as a virus category to web-blocking software, we may even be able to start claiming that the war is being won. And who knows, maybe one day the OSes that we have to work with might actually come from the manufacturer pre-secured and we could get some sleep at night. But I'm not holding my breath.

References:

Dvorak, John C. "New Generation Viruses". 1 Feb. 2001. URL:
http://www.zdnet.com/zdnn/stories/comment/0,5859,2681099,00.html (12 Feb. 2001)

Microsoft TechNet. "OFF2000: Update Available for Office 2000 UA Control Vulnerability". 26 Jan. 2001. URL:
http://support.microsoft.com/support/kb/articles/Q262/7/67.asp (12 Feb, 2001)

Symantec AntiVirus Research Center (SARC). "VBS.Davinia.B". 13 Feb, 2001. URL:
http://service1.symantec.com/sarc/sarc.nsf/html/pf/vbs.davinia.b.html (13 Feb, 2001)

McAfee AVERT Virus Library. "VBS.Davinia@MM Summary". 15 Jan, 2001 URL:
http://vil.nai.com/vil/virusSummary.asp?virus_k=98964 (13 Jan, 2001)

Microsoft product information. "Systems Management Server" URL:
http://www.microsoft.com/smsmgmt/default.asp (13 Jan, 2001)

SurfControl, Product Information. "Managing Responsible Internet Usage". URL:
http://www.surfcontrol.com/products/superscout_for_business/super_scout/index.html (13 Feb, 2001)

WinWhatWhere Corp Product Information. "WinWhatWhere Investigator". URL: http://www.winwhatwhere.com (13 Feb, 2001).

Last Updated: May 29th, 2017

Upcoming Training

Security Operations Center Summit & Training	Washington, DC	Jun 05, 2017 - Jun 12, 2017	Live Event
SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
Community SANS Ottawa SEC401	Ottawa, ON	Jun 05, 2017 - Jun 10, 2017	Community SANS
SANS San Francisco Summer 2017	San Francisco, CA	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS Charlotte 2017	Charlotte, NC	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Rocky Mountain 2017 - SEC401: Security Essentials Bootcamp Style	Denver, CO	Jun 12, 2017 - Jun 17, 2017	vLive
SANS Secure Europe 2017	Amsterdam, Netherlands	Jun 12, 2017 - Jun 20, 2017	Live Event
Community SANS Portland SEC401	Portland, OR	Jun 12, 2017 - Jun 17, 2017	Community SANS
SANS Rocky Mountain 2017	Denver, CO	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Minneapolis 2017	Minneapolis, MN	Jun 19, 2017 - Jun 24, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Cyber Defence Canberra 2017	Canberra, Australia	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS Paris 2017	Paris, France	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, Japan	Jul 05, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, Singapore	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Minneapolis SEC401	Minneapolis, MN	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Los Angeles - Long Beach 2017	Long Beach, CA	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Phoenix SEC401	Phoenix, AZ	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Munich Summer 2017	Munich, Germany	Jul 10, 2017 - Jul 15, 2017	Live Event
Mentor Session - SEC401	Ventura, CA	Jul 12, 2017 - Sep 13, 2017	Mentor
Mentor Session - SEC401	Macon, GA	Jul 12, 2017 - Aug 23, 2017	Mentor
Community SANS Colorado Springs SEC401	Colorado Springs, CO	Jul 17, 2017 - Jul 22, 2017	Community SANS
Community SANS Atlanta SEC401	Atlanta, GA	Jul 17, 2017 - Jul 22, 2017	Community SANS
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
SANSFIRE 2017 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Jul 24, 2017 - Jul 29, 2017	vLive
Community SANS Charleston SEC401	Charleston, SC	Jul 24, 2017 - Jul 29, 2017	Community SANS
Community SANS Fort Lauderdale SEC401	Fort Lauderdale, FL	Jul 31, 2017 - Aug 05, 2017	Community SANS
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event