Introducing Crunchers

Contents

1 : Introduction

A cruncher is an automation technology that takes a suspected executable and determines wether it is a virus or not. It is the "bigger picture" that has combined together suspicion, baiting, emulation, and extraction into a super virus-killing machine.

THEY EXIST, THEY EXIST NOW, AND YOU HAVE BEEN TARGETED.

2 : Proof of Life

Extract 1 : "The Hot Zone" by Srikumar S Rao

IBM isn't the only firm with new defenses against the virus spreaders. Symantec has a spider that cruises the Internet, looking at 500 known virus transmission sites and also randomly downloading files. These files are checked for viruses, using various automated analytical engines.

But then the bad guys are getting rather creative, too. Computer vandals have created polymorphic viruses that mutate each time they infect a computer, making immunization much more difficult. They have taken to encrypting viral code so it cannot be detected while inactive.

The good guys have retaliated by creating safe "virtual computers" where viruses can be tricked to deliver their payloads. They are then detected, analysed and zapped.

In a well-guarded laboratory at IBM's Hawthorne office, Jeffrey Kephart, manager of antivirus science and technology, demonstrates what the future will bring. He infects a PC with a simulated unknown virus. The protection program detects it instantly and captures the viral code, sending it securely to an analysis computer sitting a few yards away. The virus is analysed, a signature extracted and an antidote developed and sent back. Elapsed time, less than five minutes. Sometime next year IBM aims to install a system like this over the Internet to its customers.

So who's going to win this battle, the viruses of the virus hunters? That's too hard to predict, but here's a pretty safe forecast: Corporations are going to have to spend more and more money on self-defence.

Extract 2 : http://vx.netlux.org/lib/ajk01.html

At IBM, we are creating what may be thought of as an immune system for cyberspace. Just as the vertebrate immune system creates immune cells capable of fighting new pathogens within a few days of exposure, a computer immune system derives prescriptions for recognizing and removing newly encountered computer viruses within minutes. In a current prototype, PCs running IBM AntiVirus are connected by a network to a central computer that analyses viruses. A monitoring program on each PC uses a variety of heuristics based on system behavior, suspicious changes to programs, or family signatures to infer that a virus may be present. The monitoring program makes a copy of any program thought to be infected and sends it over the network to the virus-analysis machine.

On receiving a putatively infected sample, the machine sends it to another computer that acts as a digital petri dish. Software on this test machine lures the virus into infecting specially designed "decoy" programs by executing, writing to, copying and otherwise manipulating the decoys. To replicate successfully, a virus must infect programs that are used often, and so the decoy activity brings the viral code out of hiding. Other behavioral characteristics of the virus can be inferred during this phase as well.

Any decoys that have been infected can now be analysed by other components of the immune system, which will extract viral signatures and produce prescriptions for verifying and removing the virus. Typically it takes the virus analyser less than five minutes to produce such prescriptions from an infected sample. The analysis machine sends this information back to the infected client PC, which incorporates it into a permanent database of cures for known viruses. The PC is then directed to locate and remove all instances of the virus, and it is permanently protected from subsequent encounters.If the PC is connected to other machines on a local-area network, it is quite possible that the virus has invaded some of them as well. In our prototype, the new prescription is sent automatically to neighbouring machines on the network, and each machine checks itself immediately. Because computer viruses can exploit the network to multiply quickly, it seems fitting that the antidote should use a similar strategy to spread to machines that need it. By allowing the latest prescriptions to be propagated to subscribers at uninfected sites, it is possible in principle to immunize the entire PC world against an emerging virus very rapidly.

Extract 3 : http://www.symantec.com/corporate/ibm/av_tech.html

The virus research experts at SARC created what is known as the Seeker Project as a system of virus search, retrieval and analysis. The technology scours the Internet, gather viruses lingering there and create solutions for them before Symantec's customers come into contact with them. The Seeker

Project is broken down into three separate modules: Seeker, Bloodhound and SARA.

Seeker: Seeker is a Web spider designed to scour the Internet and gather files for analysis. It moves out from Symantec across the world, obtaining samples for analysis in the SARC lab.

Bloodhound: Rather than using signatures, Bloodhound uses Symantec's patented heuristic technology to detect viruses by inspecting files for virus-like behavior.

SARA: SARA (Symantec AntiVirus Research Automation) is the heart of the Seeker project. The SARA module takes a virus sample obtained using Seeker, extracts the unique qualities of the virus, develops a Symantec detection and repai scheme and tests that newly developed scheme in less than five minutes.

Extract 4 : http://www.symantec.com/avcenter/venc/data/automat.html

As part of its continuing effort to detect and eradicate computer viruses, Symantec developed Symantec AntiVirus Research Automation (SARA) technology. SARA analyses submitted files to detect new viruses and create the virus definitions used to remove them automatically.

To categorize virus definitions created by SARA, the term Automat is included in the virus name to indicate the identification method. For example, virus names contain a prefix such as W97M that describes the virus type. A SARA -detected virus might be named W97M.Automat.A. The alphabetic character suffix is applied to make the virus name unique.

SARA is fully automated. Virus analysis, definition development, and quality assurance are performed without human intervention. Once a virus definition is created, it is automatically added to Symantec's Norton AntiVirus virus definition updates. If SARA is unable to produce a signature, the submission is forwarded to Symantec engineers who perform a traditional manual analysis.

Currently, SARA is used to control the spread of Macro viruses. Because SARA eliminates numerous mundane virus identification tasks, Symantec engineers can concentrate on more difficult virus threats.

Norton AntiVirus products include a feature called Scan & Deliver that allows users to quickly and easily submit a file with a suspected or unrepairable virus to the Symantec AntiVirus Research Center (SARC). When a submission is received by SARC, the file is analysed initially by SARA.

SARA employs artificial intelligence to analyse the virus sample, replicate a potential virus, and then write a memory detection and removal definition for that virus. SARA next performs an unbiased check of its work against a rigorous standard set to either pass or fail the results.

After the new definition is tested, it is automatically sent back to the user who submitted the file. The new definition is also added to the regular virus definition updates available to all users.

Virus definitions generated by SARA are regularly reviewed by SARC engineers. After a manual review and confirmation, Automat is removed from the virus name.

Extract 5 : http://www.symantec.co.kr/press/1999/n990914a.html

Norton AntiVirus Corporate Edition 7.0 also includes a new version of the highly successful Scan and Deliver feature that provides customers with the most complete and accurate anti-virus protection cycle available.

The new Scan and Deliver is an automatic, global response process for submitting suspected or infected files and receiving new virus repair definitions over e-mail. When a new viral event is discovered at a client, the system can automatically package and forward the sample to the Quarantine Server from where it can be submitted directly to SARC. Once SARC receives the sample, it is automatically passed to the new Symantec AntiVirus Research Automation (SARA) technology which can in turn, automatically create a cure and transmit the resulting fix back to the reporting corporation. The cure is also made available to other service subscribers, which significantly minimizes the possibility of widespread infection.

This automation technology will greatly reduce the amount of time to create a cure for a new virus, providing much relief to the IT administrator from today's fast moving threats. The new Scan and Deliver is based on technology co-developed with IBM.

Extract 6 : http://www.symantec.com/press/1999/n991001.html

Symantec Corporation (Nasdaq: SYMC) today announced Striker32, the most advanced virus detection and repair technology engineered to combat the growing threat of complex 32-bit Windows-based viruses. Striker32, included in all Norton AntiVirus products, works by setting up a virtual Pentium-based Windows "clean room" in which a suspect Windows program is allowed to run. By analysing each program as it works, Striker32 is able to determine whether the program is infected. Uninfected files are processed quickly, which minimizes the impact of scanning on system performance.

Once identified by Striker32, an infected file is safely isolated using Norton AntiVirus' Quarantine feature. From there, the Scan and Deliver feature of Norton AntiVirus enables users to send the file over the Internet to the Symantec AntiVirus Research Center (SARC) for analysis and repair. Scan and Deliver includes automated macro virus analysis and repair technology that enables virus cures to be created and delivered faster than the malicious code can spread.

"Striker32 makes it possible for our researchers to analyse complex viruses such as the W32.Bolzano virus and produce cures in minutes rather than the days required by traditional anti-virus technology," said Enrique Salem, vice president of Symantec's Security and Assistance Business Unit. "With Striker32 and Scan and Deliver technologies working together, Norton AntiVirus continues to be the most advanced, responsive and sophisticated anti-virus solution available."

With Striker32, users are protected against today's most sophisticated viruses, including all 17 variants of the W32.Bolzano virus. W32.Bolzano is considered the largest family of Windows viruses .

The latest variants of W32.Bolzano have eluded detection by traditional anti-virus technology because the variants mutate and bury themselves deep within Windows executable files, hiding all signs of infection. In contrast, most traditional computer viruses attach their programming instructions to a few, well-known areas of executable files, making isolation and detection easy.

Because Striker32 has the capability to detect viruses regardless of where the virus inserts itself or how it conceals its programming instructions, users are assured of having the most advanced defense against this growing threat.

Extract 7 : http://symantec.com/avcenter/reference/striker.pdf

Like generic decryption, each time it scans a new program file, Striker loads this file into a self-contained virtual computer created from RAM. The program executes in this virtual computer as if it were running on a real computer.

However, Striker does not rely on heuristic guesses to guide decryption. Instead, it relies on virus profiles or rules that are specific to each virus, not a generic set of rules that differentiate nonvirus from virus behavior.

When scanning a new file, Striker first attempts to exclude as many viruses as possible from consideration, just as a doctor rules out the possibility of chicken pox if an examination fails to detect scabs on a patient's body.

For example, different viruses infect different executable file formats. Some infect only .COM files. Others infect only .EXE files. Some viruses infect

both. Very few infect .SYS files. As a result, as it scans an .EXE file, Striker ignores polymorphics that infect only .COM and .SYS files. If all viruses are eliminated from consideration, then the file is deemed clean. Striker closes it and advances to scan the next file.

If this preliminary scan does not rule out infection, Striker continues to run the file inside the virtual computer as long as the behavior of the suspect file is consistent with at least one known polymorphic or mutation engine.

For example, one polymorphic virus is known to perform math computations and throw away the results. A second polymorphic may never perform such calculations. Instead, it may use specific random instructions in its decryption routine. A third polymorphic may call on the operating system as it decrypts.

Striker catalogs these and nearly 500 other characteristics into each virus profile, one for each polymorphic and mutation engine.

Consider a set of generic heuristic rules that identify A, B, C, D, and E as potential virus behaviors. In contrast, a Striker profile calls for Virus 1 to execute behaviors A, B, and C. As it decrypts, Virus 2 executes behaviors A, B, and D, while Virus 3 executes behaviors B, D, and E.

If Striker observes behavior A while running a suspect file inside the virtual computer, this is consistent with viruses 1 and 2. However, it is not consistent with Virus 3. Striker eliminates Virus 3 from consideration.

The heuristic-based system must continue searching for all three viruses, however, because it observes behavior that is consistent with its generic rules.

If Striker next observes behavior B, this is consistent with viruses 1 and 2. Striker must continue scanning for these two viruses. However, the heuristics again continue to search for all three viruses. Finally, if Striker observes behavior E, this eliminates Virus 2 from consideration, and Striker now pursues a single potential virus.

The heuristic-based scanner continues to search for all three viruses. Under Striker, this process continues until the behavior of the program running inside the virtual computer is inconsistent with the behavior of any known polymorphic or mutation engine. At this point, Striker excludes all viruses from consideration.

On the other hand, a heuristic-based system scans for all viruses all the time. It must find some behavior inconsistent with all behaviors.

Clearly the first advantage to Striker's approach is speed. The profiles enable Striker to quickly exclude some polymorphic viruses and home in on others. In contrast, heuristics labor on, scanning all program files against

all available generic rules of how all known polymorphics and all known mutation engines might behave.

The profiles also enable Striker to process uninfected files quickly, minimizing impact on system performance. In contrast, heuristic-based scanning is more likely to decrease system performance, because uninfected files must also be scanned against all generic rules for how all known polymorphics and mutation engines might behave.

Second, anti-virus researchers are no longer forced to rewrite complex heuristic rules to scan for each new virus, then exhaustively test and retest to ensure they do not inadvertently miss a polymorphic the software previously detected.

Third, with Striker, a team of anti-virus researchers may work in parallel, building profiles for many new polymorphic viruses, swiftly adding each to Striker. Each profile is unique, much like a virus signature, independent of any other profile. The old profiles still work, and the new profile does not affect the old. Exhaustive, time-consuming regression testing is no longer necessary. It becomes easy to update anti-virus software by compiling new virus profiles into the Norton Antivirus database file that is posted online monthly or obtained on floppy disk.

Extract 8 : http://symantec.com/avcenter/reference/dis.tech.brief.pdf

The first step in building an automated anti-virus analysis and response system is detecting new or unknown threats at the desktop, the server, and the gateway. Suspicious files can then be forwarded for automatic analysis and processing.

1. New/unknown viruses quarantined

2. Local quarantine forwards samples to corporate quarantine

3. Corporate quarantine securely forwards samples to regional Symantec gateway

4. Gateway forwards samples to back-end automation

5. Back-end automation forwards new cure/fingerprints to gateway

6. Gateways securely forward status and fingerprints to corporate quarantine

7. Corporate quarantine forwards fingerprints to master management server

8. Master servers automatically forward samples to primary servers

9. Primary servers roll out definitions to clients

The Digital Immune System has two separate processing queues: one for corporate or government customers and one for consumers. To ensure scalability and availability, Symantec has deployed separate computer hardware to manage each of these queues, ensure that a glut of submissions on one queue will not encumber the other, and protect against denial-of-service attacks. Before processing a submission, the back-end automation adds all submission information to a tracking database.

During the automated analysis process, the Digital Immune System inserts subsequent information into a database for reference and accounting purposes. If the submission is subsequently deferred for manual analysis, all logs from the analysis and filtering steps are available to the human analyst.

THE DIGITAL IMMUNE SYSTEM FILTERS AUTOMATICALLY RESOLVE APPROXIMATELY 87% OF ALL SUBMISSIONS, WITH AN EXPECTED INCREASE TO 95% AS NEW AUTOMATED ANALYSIS MODULES ARE ADDED TO THE SYSTEM.

CASE 1: CLEAN FILE FILTERS

The Digital Immune System maintains a database of over 700,000 clean programs found on PCs running the Windows operating system. If a file in a submission matches a file in the database, the back-end system records the result in the database and eliminates the file from further consideration.

CASE 2: FALSE POSITIVE FILTERS

A false positive occurs when an anti-virus program incorrectly identifies a clean program as being infected with a virus While anti-virus companies attempt to minimize false positives, they are inevitable and commercial-grade automation processes must be able to handle such a scenario.
If an anti-virus software company distributed a new virus fingerprint that identified false positives on millions of computers around the world, it could cause a large subset of users to submit files to the Digital Immune System for analysis. Therefore, the back-end automation system must have a mechanism to identify and respond appropriately when a submission is incorrectly labeled as positive. The back-end automation system leverages a database of known false positive files to automatically identify false positives in submissions. If a file in the submission matches exactly with one in the false-positive database, the back-end system records the result in the database and excludes the file from further consideration.

CASE 3: KNOWN VIRUS FILTERING

In the final stage of submission filtering, the back-end system scans all remaining files with Norton AntiVirus, using the very latest virus definition files. The back-end attempts to repair all files detected as viruses, and if the repair is successful, IT automatically records this result in the database. This filtering step enables the back-end system to automatically resolve submissions that contain new but only recently identified viruses. Because Symantec Security Response updates its virus definition files several times per day, this filtering step can quickly identify new viruses and ensure that customers quickly get the most up-to-date cures.

ANALYSIS CENTER REPLICATION MODULES CAN REPLICATE AND COMPLETELY ANALYSE A NEW MACRO VIRUS IN APPROXIMATELY 30 MINUTES. AUTOMATIC VIRUS REPLICATION ENABLES DIGITAL IMMUNE SYSTEM COMPUTERS TO REPLICATE NEW AND UNKNOWN COMPUTER MACRO VIRUSES, CHARACTERIZE THEIR BEHAVIOR, AND AUTOMATICALLY GENERATE A CURE-ALL WITHOUT HUMAN INTERVENTION. RAPID REPLICATION AND AUTO-ANALYSIS OF NEW THREATS IS KEY TO COMBAT THREATS LIKE MELISSA. THIS IS THE FIRST SYSTEM IN THE WORLD TO PROVIDE AUTOMATIC PROTECTION AGAINST COMPUTER VIRUSES.

The Digital Immune System from Symantec deploys a back-end system analysis architecture, called the Analysis Center created by scientists at IBM's T.J. Watson Research center which offers:

1. Fast replication and auto-analysis of Word and Excel macro viruses, and of DOS viruses.
2. Multiple, simultaneous, replication and analysis sessions to support multiple customer requests.
3. Improved filtering of clean files and false positives.
4. An extendable architecture.

The Analysis Center automatically replicates and analyses DOS, Word, and Excel macro viruses. If any of the files in a submission appear to be Word documents or Excel spreadsheets, those documents are queued for processing by the Analysis Center Macro replication module. If the files contain a DOS virus, they are routed to the DOS replication module.

Symantec and IBM Research engineers are in the process of adding and rolling out additional auto-analysis modules for 32-bit Windows viruses, and for computer worms within virtual email networks such as Explore.Zip and Melissa.

The Analysis Center is a fully contained network-within-a-network. Its replication system feeds each submitted sample (e.g., a Word document) into a simulated Windows computer running on an enterprise-grade server,

and attempts to coax any viruses or worms within the document or spreadsheet to infect the virtual system.

If the document or spreadsheet contains no viruses, this will be apparent after the replication session; however, just because no viral activity was detected doesn't mean the file is not infected. Specifically, the file in question might contain a "picky" virus that fails to spread itself during the replication session. Consequently, the replication system inserts all log files and other data into the database for reference purposes, and the file is manually examined by Symantec Security Response researchers.

If the file in question does contain a computer virus, and if that virus replicates itself within the simulated environment, then the replication system gathers all potentially infected files and analyses them. If all infected files show similar infection characteristics, the replication system automatically generates a new virus fingerprint for the virus.

Next, the replication system creates a test virus fingerprint database containing the new fingerprint and launches Norton AntiVirus, with this new database, to scan the virus and all of its child infections. If the test fingerprint database correctly detects and repairs all of the infections, the Digital Immune System provisionally certifies the new fingerprint.

Finally, the back-end system obtains a copy of the latest virus fingerprint database, without the new fingerprint, and scans all of the viral samples once more. If the most recent fingerprint database fails to detect the infections, the back-end inserts the provisional fingerprint into the master database. This final check is performed to reduce the likelihood that redundant virus fingerprints will be inserted into the fingerprint database. A new set of definitions is then built including the newly created fingerprint, and automatically returned via the Digital Immune System gateway to Central Quarantine at the customer site.

Once the back-end system has replicated and analysed an infected file, and created a new fingerprint database, it initiates a house-cleaning session to check whether any currently open submissions contain the same virus; these submissions can be automatically re-filtered, eliminating the need for manual processing.

3 : General Operation

Cruncher strategy can be split into two components, the front-end that selects and submits suspicious files for processing, and the back-end which replicates samples of the virus from within the file and constructs a signature to detect those samples.

There are three front-end strategies detailed within the extracts above and we can extrapolate some ideas as to how each section works.

1. Antivirus software can send suspicious files from home/workstations if they are considered suspicious.

a) Files are suspicious if their checksums are made at install, and change.

b) Files are suspicious if their internal structure is inconsistent.

c) Files are suspicious if when run through an emulator, their code acts in rare ways that trigger heuristics, or are decrypted in a way that allows heuristics to pick up routines within it.

2. Antivirus companies can search through internet sources and submit them for testing.

d) Searches are conducted on newsgroups and websites (particularly hacking, warez and virus sites), and all files are sent for testing.

3. Users can manually submit suspicious files for processing.

e) If files are sent through email or existing files run noticeably slower, then users are likely to send them for testing.

Once selected, the files are processed by the back-end using these strategies.

1. Files are checked against filter strategies to reduce time wasted during an outbreak.

a) A database of known clean files is kept, and compared against submitted files before any other processing is done. This is most likely the same software database used to exclude false-positives when extracting virus signatures.

b) A smaller database of previously known false-positives is kept, and when any are found, newer signature files are sent to the client.

c) A scan is done using latest in-house signatures to exclude very recently detected viruses (ie: in the past few hours).

2. Proper analysis begins.

a) Files are moved into a virtual computer (and virtual network) where they are forced to reproduce by different modules. These modules most likely execute macro programs through set steps, or programs in set directories containing baits, or opening and forwarding dummy emails.

b) Once samples (most likely files copied from the false positive database) have been modified from their original forms, an emulation is run to get an unencrypted form of the virus, and signatures tested against samples, to see if they are effective.

Bait technology can be implemented by crunchers in a few ways. Either they implement a system that generates random filenames with varying sizes and runs them through various tests trying to get them infected. Or they can simply set up a full "normal" installation of a system and use scripts to run programs normally, hoping that some get infected. Or they can just fill a directory with a database of real program executables and watch what changes.

4 : Defensive Strategies

If is a well publicised fact that thousands of viruses have been detected on web sites by Seeker, and had their signatures extracted long before ever having a chance to replicate in the wild. Your best defence against this is common sense, don't publish malware in ways that automated engines can find the files and extract the contents.

Depending on the method used to compare files against the known clean database (either various checksums or full file comparisons), interchangeable checksum neutral code may fool the cruncher into bypassing some infections, creating an incomplete detection signature (or none at all).

Furthermore, the internal structure of files should be made more consistent so that headers arouse less suspicion. Anti-emulation code can be tested against antivirus engines by wrapping known viruses in polymorphic decryptors, so that signature extraction by the cruncher simply will not find any matches.

Let it also be noted that code must not rely on condition tests of any sort as emulation systems often follow bad-condition paths just to see what happens (a good example of this is described in the Striker32 paper of which extracts are given above).

Gauge system parameters like disk and keyboard activity, and watch for changes before doing infections. Keep an internal clock and wait before infection, so that automated systems running on a short fuse can't virus get samples.

Baits can be avoided by stepping outside of the same-directory mind frame, and focusing on directory structures, program methodology (is it in the registry, does it import support files), and even viewing networks as a whole entity, by comparing installations across accessible shares and doing web searches on a filename to see if it returns matches.

5 : Conclusion

Cruncher technology is a closely guarded secret in antivirus firms, because of the substantial research investment and the possibility that exposing too much information could give the leg up to competitors or teach virus writers how to cheat the system.

You'll notice that this article is mostly Symantec based, and this is because they are the only company that markets to the enterprise using information

about the technical expertise of their internal processes. But it should be clear that other antivirus companies will have similar, but secret, processes.

While it would be interesting to gain further information about the specific vulnerabilities of the Cruncher systems in use around the world, it is likely to be far less valuable than thinking in general terms of what is, and what is not, a good idea in your virus activities.

In 1998 we were talking internally "If we were asked to create a Cruncher, how would we go about it?" Funnily enough, everything we discussed, from ruling out submissions with a software database, and automated bait generators with signature extraction, became reality in the Symantec project.

So the most important lesson learned here is that virus writers can be just as intelligent as antivirus programmers, and that it is possible and fruitful to predict (and counter) future antivirus technology before it is implemented.