

Trace-Back

A Concept for Tracing and Profiling Malicious Computer Attackers

Ofir Arkin
ofir@atstake.com

Atstake Limited
6-8 James St.
London, England
www.atstake.com



In memory of victims of terrorism

Introduction

In the computer security arena, every now and then, a vulnerability comes along causing a significant impact. The impact of a vulnerability is based on factors such as popularity of the vulnerable platform and the ease of exploitation of the vulnerability. Lots of research gets done on a vulnerability, beginning from its origin to the various permutations and combinations of exploit code that come out subsequently. In recent years, we have seen self-propagating exploit code (in other words, worms) becoming quite popular.

Very little is known about the events taking place in the time period between the instance that a vulnerability gets discovered by an individual or a small group of individuals, and the moment when exploit code becomes publicly available on the Internet. To zero in on the origins of a particular piece of exploit code is quite a daunting task. Very little research has been done on the subject outside of government or military organizations. Tracing back origins is a very tricky task, especially if one has to reconstruct events backwards. This paper addresses this very issue - trying to roll the film reel backwards from the time the exploit code becomes widespread in public, and filling in the blank frames to the beginning of the movie. This may not be the ultimate "big-bang" theory of the exploit universe, but it provides us with new viewpoints on exploits and their originators.

One of the problems the computer security industry and law enforcement agencies face is tracking and tracing back malicious computer attacks to their origin, and associating the attacks with real individuals. Malicious computer attackers may use several techniques to fool tracers by disguising their real location, so they cannot be traced back and prosecuted. A highly skilled malicious computer attacker can "get away with it" with the technology of today when combined with the lack of skills, methodologies, resources and time available to the law enforcement agencies.

So how can technology overcome the obstacles and provide the law enforcement agencies with the tools and methodologies needed to trace the real attackers?

Using monitoring, intelligence gathering, and electronic surveillance systems that are operational today by many intelligence agencies¹, sometimes in a joint effort, we are able to locate, precisely, exploits and exploit code and bind them to the originator of that exploit with a high degree of accuracy. Not only are we able to determine the virtual identity of the code writer / attack generator but also to track her real identity in real life.

Looking back at the September 11th tragic events and examining publications in the past, it is known and published that several intelligence agencies maintain the ability to electronically gather and store any communications activity whether it is on the Internet or done by any other electronic means². It is not known for how long this communication intelligence is being stored for, and what exactly is being logged although we can imagine that the logs contain everything these agencies can receive (and retrieve).

The data collected is not only good for locating individuals that have committed crimes in real life by using some of the Internet's abilities. A good example of this is using the logged information to track airline ticket buyers that were using the Internet. The data stored can also be used to track malicious computer attackers / black hats with an amazing degree of accuracy, as I will demonstrate.

So, how is it possible to trace back these malicious computer attackers?

Two methods are presented in this paper.

¹ FBI Congressional Statements , [Carnivore Diagnostic Tool](#), Donald M. Kerr, Assistant Director, Laboratory Division, before the United States Senate, Committee on the Judiciary, 09/06/2000: <http://www.fbi.gov/congress/congress00.htm>.

² The European Union investigation into Echelon: http://www.europarl.eu.int/committees/echelon_home.htm. Meet the FED panel, Defcon 8: <http://www.defcon.org/html/defcon-8-post.html>.

The First Method - Looking for a needle in a haystack

It is all a matter of patience, and not so advanced technology and techniques that are based on tracing events which have occurred in the past.

We are going to refer to two time frames: The recent past, and the distant past.

Recent Past

In the recent past we can identify, in real time or in real near time, an attack attempt.

How can we differentiate between attack attempts and regular traffic if the attacks are not known?

By determining what traffic is legitimate and what is not legitimate. According to patterns in applications, for example, or patterns in shell code³ or the shell code encode as another example, IPv4 legitimate traffic and other means.

In terms of Intrusion Detection Systems what we are going to observe in the recent past is the exploit's signature.

What can we do with it?

We can look at the stored information for the same exploit code signature, and trace it back as long as we can.

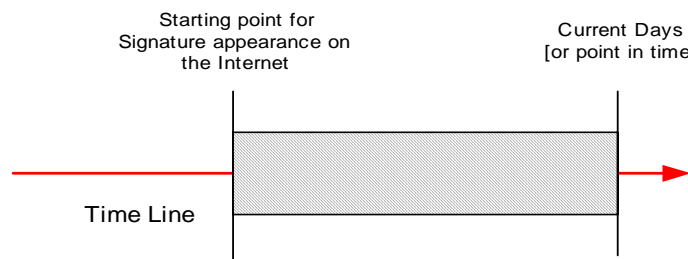


Figure 1

We can draw borders in time between the period that network signatures for the exploit we look for started to appear on the Internet to current time or any period of time we wish.

Tracking the activity on some mailing lists can, sometimes, give the knowledge of some of this time line, but not all of it.

We can divide this time frame, or window frame, into one more window.

³ The term "shell code" comes from the assembly language code that is appended at the end of a long string of characters that is used to generate a buffer overflow situation. When a buffer overflow takes place within a process' memory space, the exploit is designed to send a specially crafted buffer which executes the machine code appended to it. In most exploits, this code ends up spawning a shell on the system attacked, and hence the term "shell code" has become associated with buffer strings containing machine code. The term "egg" has also been associated with such buffer strings (definition made by Saumil Shah).

Trace-back: A Concept for Tracing and Profiling Malicious Computer Attackers

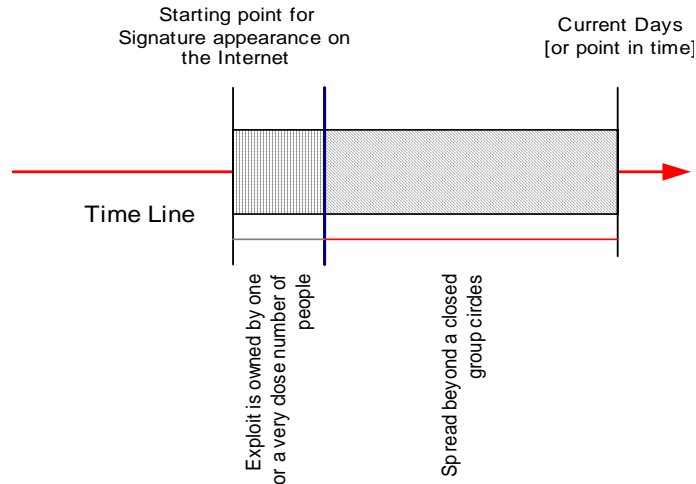


Figure 2

The extra window is the time that the exploit was shared among a small number of people. Sometimes it is owned solely by the exploit writer for a period of time (sure I could have added another window for that). As we go further back in the time window which we saw network signatures that match with the exploit, we are also able to reduce the number of people that have shared the exploit.

When the exploit signatures just started to appear it is the time that our window can be narrowed sufficiently and be used to conclude that the exploit writer (or some one he gave it to directly, or sold the exploit to) is responsible for the damage that the exploit caused in that period of time, and for the activities of it. There is always a narrow window in time in which an exploit was shared among a small group of individuals and/or is being tested by a research group against public machines.

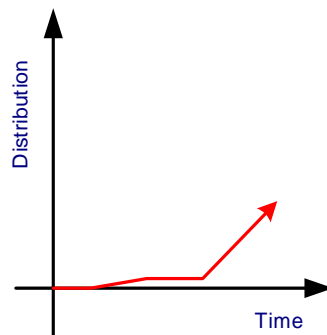


Figure 3

Who owns the exploit in this small window of time?

Sometimes, because of the nature and level of damage that an exploit might cause it is held only by its writer. Sometimes the writer offers the exploit for sale to criminal entities and/or commercial

entities and/or government agencies⁴. It is all a matter of price and gain. If the exploit writer is a member of some group she might share this exploit among her friends.

Usually the exploit code gets leaked and sooner or later the underground computer security community gets hold of that exploit code (and eventually other people as well). Or the knowledge of the vulnerability existence leaks and then exploit code is created by others. This is also the time that, usually, there is an increase in damage caused by the use of that exploit, and more and more damage reports are being collected from system administrators and systems such as distributed IDS systems and the HoneyNet⁵ project.

We might face a slightly different scheme of distribution (and ownership) when, for example, the exploit code writer faces some problems during the coding stage or in the initial testing phase. For example when the code writer has troubles making some component of the exploit work correctly and turns the code over to someone with better skills to fix, finish or help.

On the other hand there are certainly other scenarios where the exploit code will not be shared. For example if the author of the exploit possesses higher coding skills or understanding in a certain field of technology.

The time which an exploit will only be owned by one person or by a small group of people depends not only on the gain that person or group might have from using the exploit, but also from the psychology of that women or the group dynamics (fame, glory, ignorance, control). Eventually the exploit code will be leaked out (a good example is the `telnetd` exploit from the hacker group `teso`⁶).

We have identified the time frame in which the usage of the exploit increased. We also identified a small time frame, usually at the beginning of the larger time frame in which a small group of people were the only ones who shared this exploit code tested, perfected and then utilized in a hostile manner.

The early stages in the exploit code evolution and distribution is our key to solve the problem.

Using this information we can also identify IP addresses of hosts that issued the attacks. Usually they will be compromised hosts, which the malicious computer attackers broke into earlier in time and were used as launch pads for different attacks. We do not expect our attackers to connect directly to these hosts; they might use a chain of compromised hosts to hide their activities.

So how can we advance from this point on?

Distant Past

You still need to remember that we are dealing here with huge amounts of information and logs taken at different times.

In recent days, or in the near future, someone from an intelligence agency, intelligence community or from a law enforcement agency will get hold of the exploit code itself.

Until this point we were relying on signatures of the exploit, but nothing more than that.

Since we are now holding the actual code of the exploit we can search our database and logs for any *code* transfer from any given point using any electronic means of transport (protocol wise –

⁴ Evidence gathered from the Underground community.

⁵ <http://project.honeynet.org>

⁶ <http://www.securityfocus.com/bid/3064>

Trace-back: A Concept for Tracing and Profiling Malicious Computer Attackers

i.e. SMTP, FTP, SSH or SSL). This means that our signature now will be the actual exploit code (and not the exploit networking signature).

We are not limited for searching the exploit code in the form of a precompiled code only. We can also look for the exploit code in a binary form, compressed form, tared form, gzipped form, etc.

We will also have to consider that the exploit author had to test their code and perfect it in a live environment (in some cases it may take a lot of iterations until the exploit code will be good enough to go to "production"). In these instances some black hats may not wish to have their code leak so will potentially launch their attacks against known hosts / friendly hosts (i.e. my friend has this SUN Solaris box – I'll own him, it'll be a bit of a laugh and he'll understand) from their own machines. It is, sometimes, much more likely in this stage of development, they will use a host that has some connection to them – their school, their mate's, their company's server... This is, sometimes, how you get the lead that can assist in narrowing down not only their geo-location but also their network of compatriots.

In these situations the code used in the developing exploit may not differ a great deal from the end product, so using all the stored network traffic we should be able to pinpoint the attacker testing the exploit against these known hosts. In these instances one observes a situation depicted by the illustration (Figure 4) shown below⁷.

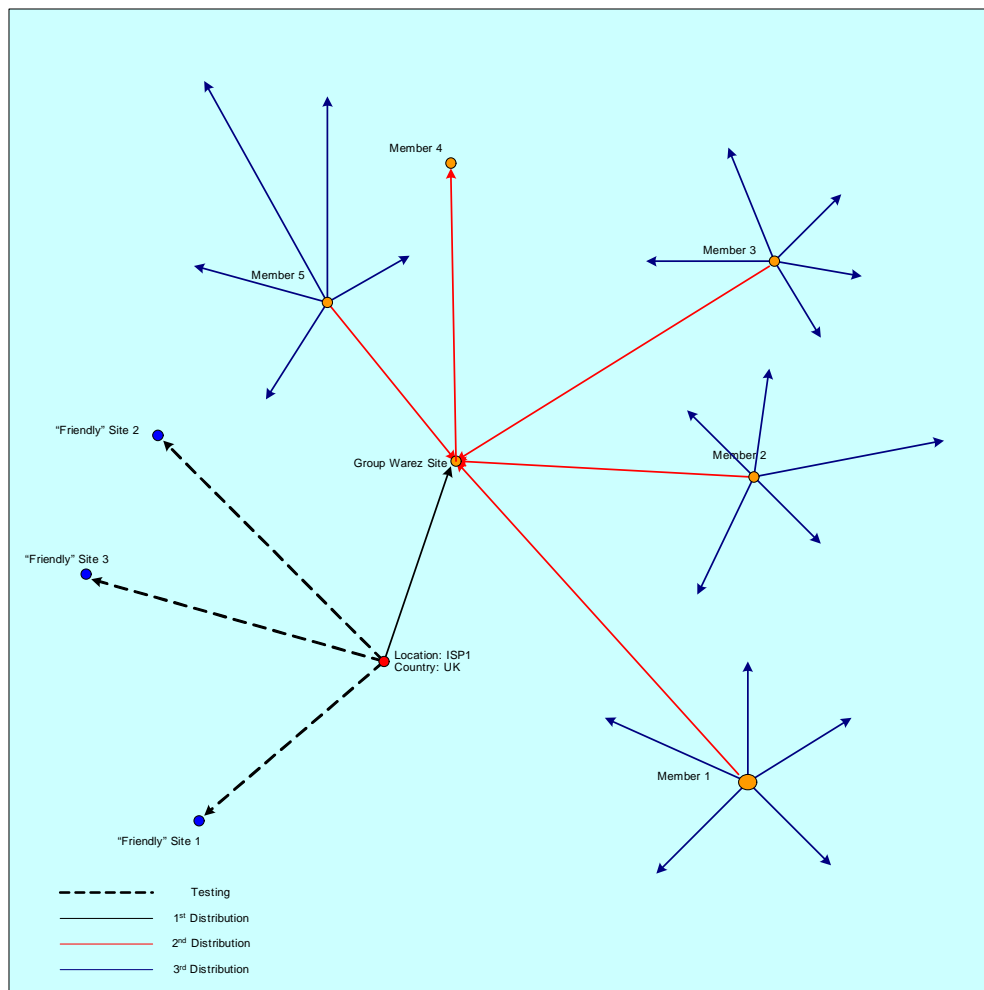


Figure 4

⁷ Idea contributed by Ollie Whitehouse [ollie@atstake.com]

Trace-back: A Concept for Tracing and Profiling Malicious Computer Attackers

With this example, you see initially very small attacks originating from one network against what we will term “friendly hosts”; we then see code with a signature matching this exploit being transferred using the File Transfer Protocol (FTP) from the author to the group’s warez server. Once there we see five downloads to separate networks all over the world. From this we can see that this historic data has become invaluable in watching the original propagation of the exploit. You may even find where the exploit leaked to the public domain if you keep digging long enough.

In addition we can start looking for any code uploads to the compromised hosts that were used to launch the exploits, and trace back from that point in time.

This means that if the writer of the malicious code transferred her code to different hosts until she reached her real launch pad, we will be able, looking at the logged transactions, to identify which hosts were used to bounce through until we reach the attacker’s real workstation on the end of a leased line, dial-up or DSL line.

The narrow window of opportunity when the exploit code was starting to be used in the wild gives the tracers even more opportunity to locate exactly the source of the exploit code, and its creator. It gives the people who are performing the trace the IPs to start with and then start tracing back and to exactly locate these malicious attackers in the huge pile of information they might have.

The time taken to spread and upload the exploit to different machines and the time we have started to see the exploit being used in the wild might overlap partially as well.

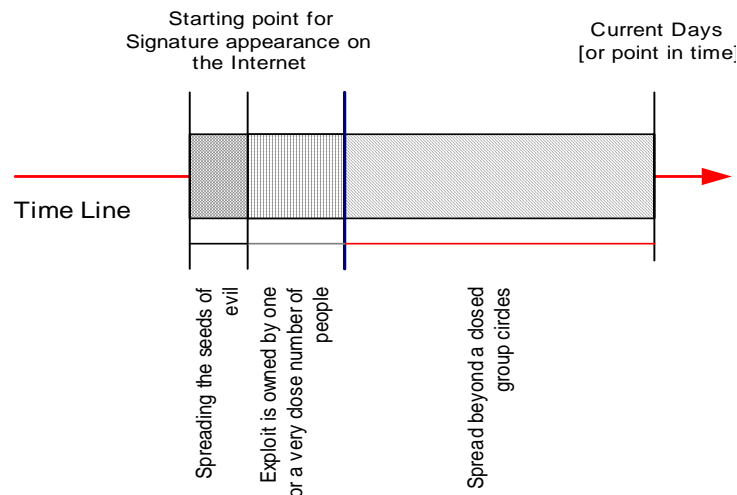


Figure 5

The time frame keeps moving backwards but our window of opportunity only increases.

Eventually, because of the vast amount of information stored in today’s surveillance systems, these systems will be able to locate a malicious computer attacker, back to it’s point of origin.

People may be skeptical about features and amount of information that a system like this can store. If you have watched the movie “Enemy of the State” (1998) starring Gene Hackman and Will Smith and you did not believe the things you saw in that movie three years ago, I suggest you go and read the European Union’s report on Echelon, and the FBI’s press releases and congressional public reports of Carnivore, as well as other publications.

What will happen if the exploit was written in a country that has no surveillance mechanisms in place? Any virtual crossings of the electronic boundaries of these countries onto the networks of the countries that does impose this kind of surveillance technology will “show up on the screens” of those monitoring systems. We might miss several pieces of the puzzle since some early propagation might occur inside that country, but the origin of the exploit code and the source for the trouble will be revealed.

It is interesting to understand what that country's pipe to the Internet is and how this connection is made. If we take satellite communications for example, it is long known that this type of communications is being monitored.

As an example one might choose a county and try to map its global crossings. A simple way to do so is to use the `traceroute` or `tracert` programs against the IP addresses of that country's biggest ISPs. These ISPs usually will use high bandwidth connections to the Internet backbone, connecting that ISP with one or more major Internet crossings. By mapping these Internet crossings we can speculate who is potentially “listening” at the other end. You need to use different Internet locations outside of the country you have chosen to check this against.

One might be surprised to notice that most of the Internet traffic goes through numerous connection points only.

Here we are getting into another area of profiling exploit code writers and malicious computer attackers. This can be done according to several keys that will be outlined in the profiling section of this paper.

Resistance is futile...

There are several methods which malicious computer attackers might use in order to try to circumvent some of the modern surveillance systems. But will they succeed?

Stenography might be used as a facility to hide code in images when transferring the malicious code between hosts. But will it be too strange to see a certain picture being uploaded to several hosts? Even if the pictures are replaced between each and every upload, a good profiling system (as part of the big surveillance systems) should pick this up and state that pictures (jpeg, gif, bmp and so on) are being used as vehicles to carry malicious code (remember the last host in the chain? The one that is being used as the launch pad for the attack?). Not mentioning researchers looking at profiling picture files that are being used for stenography...

If we examine the source code or reverse engineer⁸ the binary of stenography based programs we suspect to be used by malicious parties, we might be able to identify the usage of these programs when examining suspicious pictures.

Even if physically uploading the malicious code to a host and then connecting remotely is not that helpful for the malicious computer attacker, even if the system the malicious code was uploaded from is loaded with users from all over the Internet. If that system is being used as a launch pad for executing the code, sometimes one needs special privileges on the machines in order to launch it, and so on.

Think again, the surveillance systems are watching. Add a very good profiling capability and almost nothing will get past the radar...

⁸ Reverse Engineering is the process in which one takes a compiled program, called a binary, and uses tools to learn what was the code it was compiled from.

Profiling⁹

Criminologists and police forces use profiling techniques to help catch criminals, usually the worst kind. Using profiling techniques in solving, and tracking computer crimes is also helpful. The means to do so are totally different in terms of the technology used from those used in real-life crimes.

Adding a good profiling system to a modern surveillance system adds additional power and abilities to it.

Correlated attacks (as an example) or the usage of multiple hosts, which were compromised or not, against a common target are discovered as a result of nothing more than pure luck and/or guessing. The ability to gain an understanding that an attacker is using more than one source/host in order to gather information and/or attack a certain site is lacking from main stream distributed IDS systems.

How can someone overcome some of the problems associated with these types of attacks, which are the least reported ones? This is typically either due to the fact that it's a very slow scan/information gathering exercise and thus just looks like Internet noise or the fact that the network in question has no means of detecting this type of attack due to their use of signature based IDS systems.

Using a combination of Passive Fingerprinting and Psychology – Profiling (utilizing both technology and psychology to allow you to build the profile).

We know that one human is different from another (except when cloning occurs but even in instances of cloning the nature/nurture arguments come into play). This is a fact that no one can argue about. However, humans often follow behavioral patterns that are predictable by theory (this is due to the fact we are creatures of habit). Let's try to reflect that on the computer security field and especially on malicious computer attackers. Most of the malicious computer attackers have their own style of doing things (talking about highly skilled black hats and not script kiddies). There are a certain proportion of behaviors that an individual may exhibit that are idiosyncratic patterns that are usually unique to an individual. This is the place where psychology and social psychology comes into play. What if we can identify a style or an operational pattern? (And what if we can guide its actions...?) I can give an example of black hats that have monetary motives; other might have a political agenda and so on.

Motivations can be of considerable assistance in understanding the nature of exploit events. Many may remember the old motivations acronym used by counterintelligence agencies for years - MICE, which stood for **M**oney, **I**deology, **C**ompromise and **E**go. An adaptation of this acronym – MEECES, can be applied to the motivations driving computer exploits. In short, MEECES stands for **M**oney, **E**go, **E**ntertainment, **C**ause (basically ideology), **E**ntrance to a social group, and **S**tatus. Each of these motivations plays a role in shaping the behavior of the objectives of the exploit, the selection of means by which the exploit works, the types of hosts that are targeted, etc.¹⁰

Will this mixture of technical and psychological profiling allow us to predict and guide a certain unknown individual's actions? Will it allow us in the future when the same pattern of behavior (and of course some technical necessities combined with it as well) will pop up to mark a certain individual in future incidents and attacks? I believe that we can track this over time...

Just think about the information a modern surveillance system holds, tracks and stores.

⁹ I would like to thank Max Kilger and Jeff Stutzman for their comments for this section.

¹⁰ Contributed by Max Kilger.

We should also not forget the deterrence component of this kind of profiling. By letting people know in general terms (without revealing the secret recipes) that there are profiles built over time and that eventually they result in apprehension, the psychological feelings of risk increase for the author for each new exploit he or she writes. At some point the author encounters a threshold at which they make the decision that "the odds are against them" and they decide to curtail their activities. It does not work for everyone but works for a large percentage of people...

Usually the way to discover a correlated attack is to look at the type of queries launched from several, usually unrelated IP addresses against a common victim site or network (sometimes from geographically disparate regions). It can be an information gathering exercise or even exploitation attempts. This works for the obvious and most common things. For example we can name information gathering such as scanning, especially port scanning. One malicious IP address (host) scans port 21 while another host tries to connect to port 53. Usually the time gaps between the different attempts will be short, and the tool used will be the same one. The picture gets unclear whenever the time frame gets longer. How then can you correlate attempts? Is it pure luck? Expertise of the people who look at the data? Different processing of time periods on the collected data? Craftware? Sorcery? What happens if the targeted site is a high profile one? What will happen then? Black hats malicious attempts might be mangled with script kiddies stuff. How do you really understand what is going on?

Also remember that in some cases black hat attackers will handover the compromised host to a script kiddie once their objective has been achieved. This will allow the script kiddies to divert the heat away from the attacker as they will typically fill up the logs which will produce a good enough case against the kiddies in question (and if the black hat is lucky they may even start rm'ing parts of the system to allow them to draw attention to themselves).

In my own humble opinion you don't have the understandings of exactly what is going on. There is simply too much out there to process. Sometimes you try to process the logs (if you save them for further processing) in different manners so you might hit something after it happened. Usually it is never on a real time basis. However this must never be discounted as for a forensics exercise these types of logs are invaluable.

Another approach that might help the security community is a very simple one: let the legitimate traffic go by, mark the non-legitimate. Process them in a different manner. You can use it the way you like. For example with Firewalls you let the legitimate traffic in and drop the non-legitimate (As an example we might look at a TCP packet which both the SYN and the FIN flags are set). With Intrusion Detection systems you process the non-legitimate traffic differently than legitimate traffic. It is also a good idea to look for new problems or new exploits in the non-legitimate or unusual traffic. The definition of non-legitimate and unusual is still out of the scope of this paper due to the complex methodologies behind understanding what traffic is considered normal, this increases in complexity when we look at protocols such as SOAP which balance on top of HTTP.

What we need is to approach this problem from several perspectives. We need to isolate the unknown from the obvious. Usually exploits will belong to the non-legitimate or the unusual traffic that we might see.

So we see a new attempt / exploit out. From it, how are we able to learn more about the people who are using it?

Usually the 0-day¹¹ exploit will be owned by a small number of people. Sometimes by one individual. From looking at unusual traffic gathered by different sensors on different networks we

¹¹ Defined as computer language code written to take advantage of a particular vulnerability which has been discovered but is not publicly known.

can get a picture of hosts which are under the direct control of that individual or group of people and the nature of the operating systems they are using to launch their attacks from, as well as the nature of the operating system the exploit was written on (usually a favorite operating system is a holy thing among black hats, as well as development platforms).

The passive operating system information we gather will help us to profile the attackers according to their code writing style. Other questions like: did the attacker spend a lot of time writing the hack? Might not be answered, sometimes, although skilled coders recognize the ease or difficulty in certain exploits and can evaluate the difficulty and therefore skill level (and time involvement to some degree) necessary to have authored the code.

We can look at a number of parameters inside the exploit code

Did the code writer use a borrowed portion of the networking code from someone else, or from another tool (i.e. how many exploits have been seen reworked from LSD shell code)? What are the IP header fields the attacker is introducing and with what parameters? Which IP header fields share the same parameters again and again?

We can look at a number of parameters drawn from the network traffic we see

Did the individual launch an information-gathering attempt beforehand? Did the exploit succeed in its aim? If not did the attacker try to knock over the attacked host with whatever she had in her arsenal? Did she show frustration? Panic? Anger?

We can look for several patterns after a successful compromise

What happened after the compromise occurred? Did you see the usual script kiddies rootkit being used or was it a simple log clean and go? What was the rootkit uploaded after the compromise? Was it a downloadable root kit (such as the Linux rootkit)? Or was it the personal signature of a black hat that's using her own rootkit? Did she upload some kind of a program to the compromised host which is aimed at encrypting her future communications with the compromised host?

We can look for patterns in the way the aftermath is being done

Is this seems as an automated process? Some black hats are using programs that are not only exploiting a targeted host, but also have capabilities to automatically cover the tracks of their master and hide his presence in the compromised host.

If this code is being shared among a group of people (which may or may not belong to the same group) according to behavioral patterns within the group we might actually differentiate between certain individuals within that group as well.

A Honeynet usually uses only one machine or a low number of machines acting as honeypots, eyes and ears. What if this kind of technology/thinking could be used by an entity that has multiple sensors across continents...?

This suggests that governments with a global reach, who will deploy global Honeynet networks, will have the ability, not only to get 0-day exploits before anybody else, but will also have the ability to track down the 0-day writers faster and closer to their origin.

These governments will have the ability to use the 0-day exploit code as they please. This suggests the usage of the 0-day exploit code against other servers in other countries before patches are created for this type of exploit (a nice type of Information Warfare), or they also might send the 0-day exploit to a software company which its servers or technologies are being exploited so this problem will be fixed faster.

But bear in mind that fixing the vulnerability faster than the propagation of the exploit code may take place, might tip the exploit writer that something is wrong and probably they are being monitored. Especially if the code is owned only by one person, which is the coder of the exploit.

In cases where some puzzle pieces are missing from the bigger picture, profiling the activities around the exploitation attempts, code testing, and code distribution might lead us to certain individuals with certain behavioral patterns.

In the future when these individuals will not be as careful...

Profiling and reducing the amount of information logged

It is known that several capabilities allow an intelligence agency to record a conversation if certain words are mentioned during the conversation. What if a similar technique could be applied to computer code / network traffic? What if we are able to log network traffic and look at suspicious or predefined patterns and only if we find them interesting we will keep this "conversation"? Or even flag the two IP source and destination addresses as suspicious and try to digitally perform surveillance on that pair of IPs or set of IPs?

The Database behind the scene

What we wish to have is not just a "flat file" database of signatures but a relational database that allows you to do a number of things including:

- Relate commonalities like common code segments among different exploits
- Build profiles of exploit writers in terms of method of attack, favorite types of hosts to attack or takeover in assisting an attack (might be regional or geographical), etc.
- Psychological profile details that suggest motivation, IRC channels they might hang out on that should be scanned, potential related websites, etc.

The Second Method - Flagging and Tracing in near Real-Time

If the first method was a hard one to digest, the second method is a very simple one to understand.

Usually hosts used by malicious computer attackers will be compromised hosts, which the attackers broke into earlier in time. They will usually be used as launch pads for different attacks, or to hide the tracks of the attackers (profile the client they are using to connect to their compromised hosts... can be even different encryption schemes, etc.). We do not expect our attackers to connect directly to these hosts; they might use a chain of compromised hosts to try to hide their activities.

Malicious computer attackers might need a large number of compromised hosts at their disposal. In order to do so they might use a malicious program that will scan for known vulnerabilities, or for vulnerabilities only known to them, and for known operating system platforms vulnerable to these vulnerabilities and then exploit them.

With a global surveillance system we are able to tackle this problem in two ways:

- Flag already known exploit attempts
- Flag abnormalities

Flag already known exploit attempts

With this method using a global surveillance system we will be able to look for known exploits and their respective network signatures. This means that we will be able to flag a lot of hosts that are, and were, compromised using those known exploits (and of course the origins of the compromise).

Since hosts used by malicious computer attackers are, usually, compromised hosts, which the malicious computer attacker broke into earlier in time, keeping an eye on the compromised hosts we identify will, eventually, lead us to absorb a new exploit either being used or uploaded to one of these compromised hosts. If we are lucky enough then this exploit will be uploaded, somehow, to one of these compromised hosts, and from there either used or distribute itself to other machines controlled by the same malicious computer attacker. If we monitor all the systems that this particular malicious computer attacker used, we will then also be able to track him as well.

On occasions we will not be that lucky and monitor all the systems that a malicious computer attacker controls. But still, we will be able to see some of the distribution chain of the exploit, and might even track down a member of the group downloading this (or other) exploit from a group controlled compromised host.

As a result script kiddies will be caught like flowers after the rain.

What if the malicious code that is being uploaded is encrypted?

We still are able to identify abnormalities in the traffic generated. Either the encrypted tunnels used by the hacker to distribute his code, or the traces the exploit code creates when used from an already compromised host against another host on the Internet. Still it gives us a good start as for what to monitor and what we might expect. This also will contribute to the profiling process.

With the next example illustrated in the next page (Figure 6), a global surveillance sensor system has identified a known exploit being used by identifying its network signature. The tracking system will automatically start to track the initiator of this attack and his mark. The attack was

Trace-back: A Concept for Tracing and Profiling Malicious Computer Attackers

successful and following was a successful login attempt from the attacking host to the mark (sometimes the attacking system and the system used for remote login will be different).

Several days later the attacker using the `telnet` command, logs in his compromised host and downloads some tools and exploits over the Internet from another host (this can also be done in an automatic manner when using a Master-Slave module).

Days later, the compromised host is being used as a launch pad to attack another host. But this time the exploit code used is unknown to the surveillance system's exploit signature database, and therefore it is being tagged with priority (we can look for similar network signatures in our not-identified network signatures database entries). If the surveillance system will go back to the logged information from several days before, it will be able to extract the new exploit out of the traffic exchanged between the first host compromised and the FTP server. It will be then able to understand the exploit targets and its way of operation.

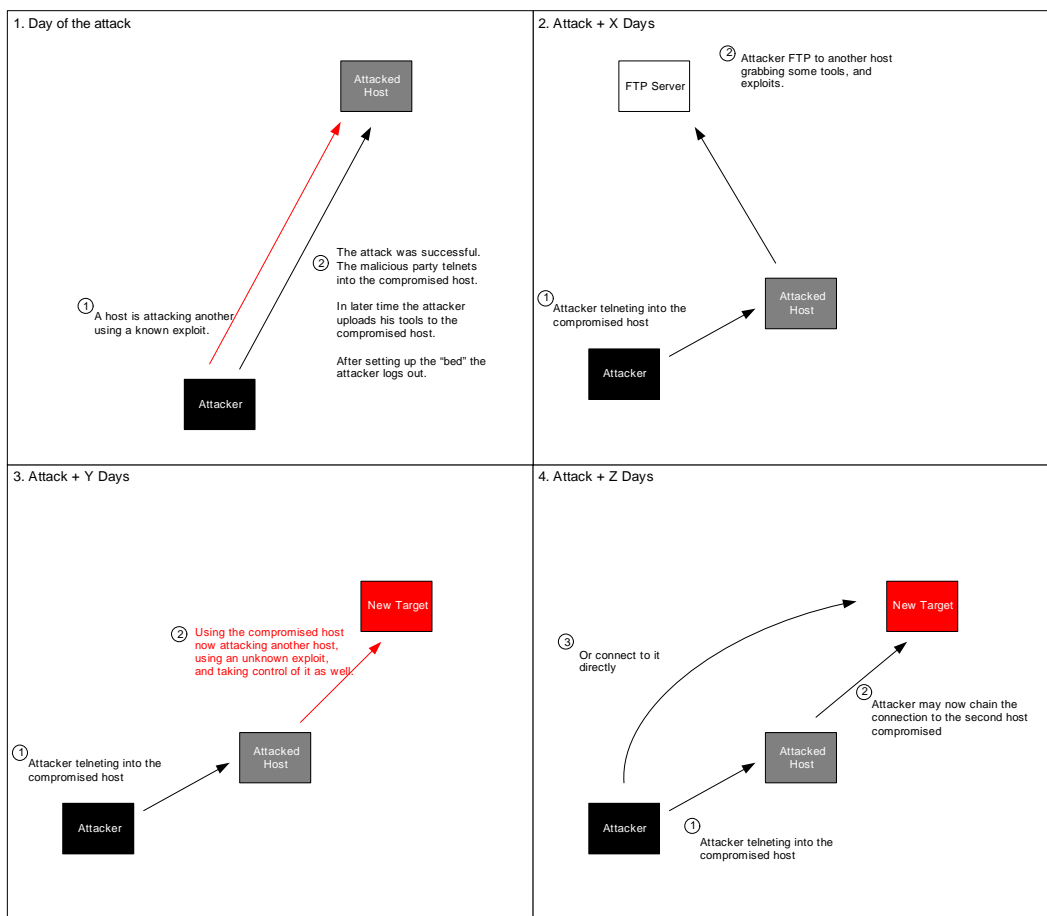


Figure 6

During the tracking period the global sensor system might be ordered to monitor the connecting hosts to the host which initiated the known exploit in step 1. It might give other details about the host used, and might lead to a step forward in tracking the real attacker.

There are, also, other meanings and ways to use the capabilities of a global surveillance system of this type, which this paper does not intend to cover.

Flag Abnormalities

With the second iteration of this method, we are looking at traffic which is non-legitimate. We can usually identify “pockets” of this kind of traffic. Sometimes the cause for the non-legitimate traffic is badly built applications and hardware. But as time passes, we will be able to identify their common patterns and give them less priority over other non-legitimate traffic.

What we might observe is something similar to the next diagram:

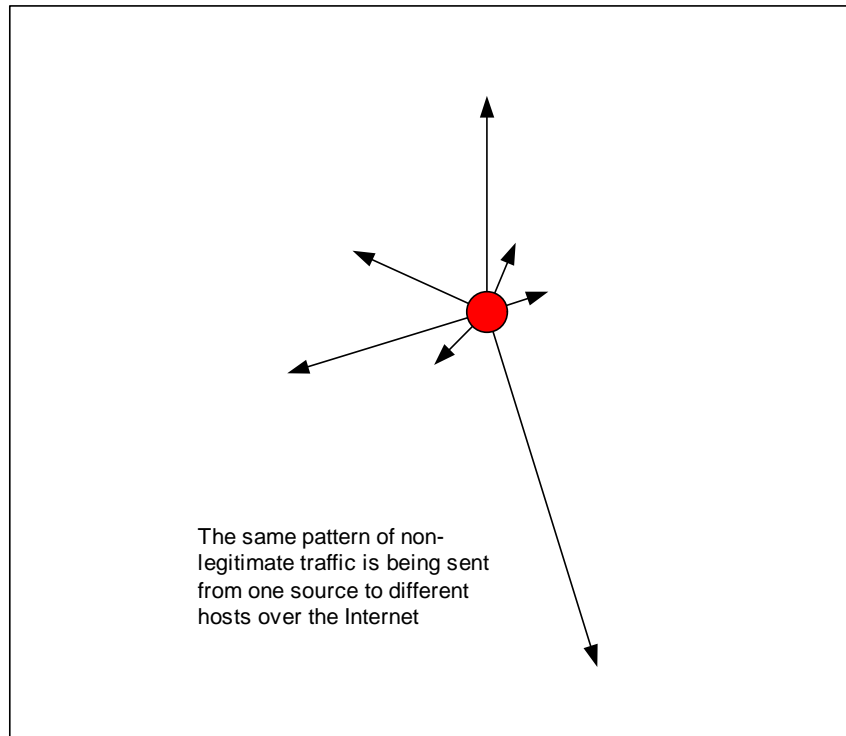


Figure 7

With this example we see the same host sending non-legitimate traffic to a number of targets over the Internet. The patterns of the traffic are similar or nearly similar. Sometimes the host will initiate several traffic flows carrying very similar traffic to the same targeted host. This might be the case where an attacker tests her code and needs to tune it.

The type of systems the exploit will be used against will have several similarities, usually the same operating system being used. This is when a new exploit is targeting an operating system and not a particular application. The targeted operating systems will have to share a common open port, and the ability to remotely login into, in order to examine the “tests” results.

Since this is a testing phase, the attacker might be using her own home system for testing. If not, she could be using another system that she trusts its security, so her new exploit being developed will not be stolen away. This gives the trackers a better chance in detecting the code writer and tracing her back.

Let's examine the next case:

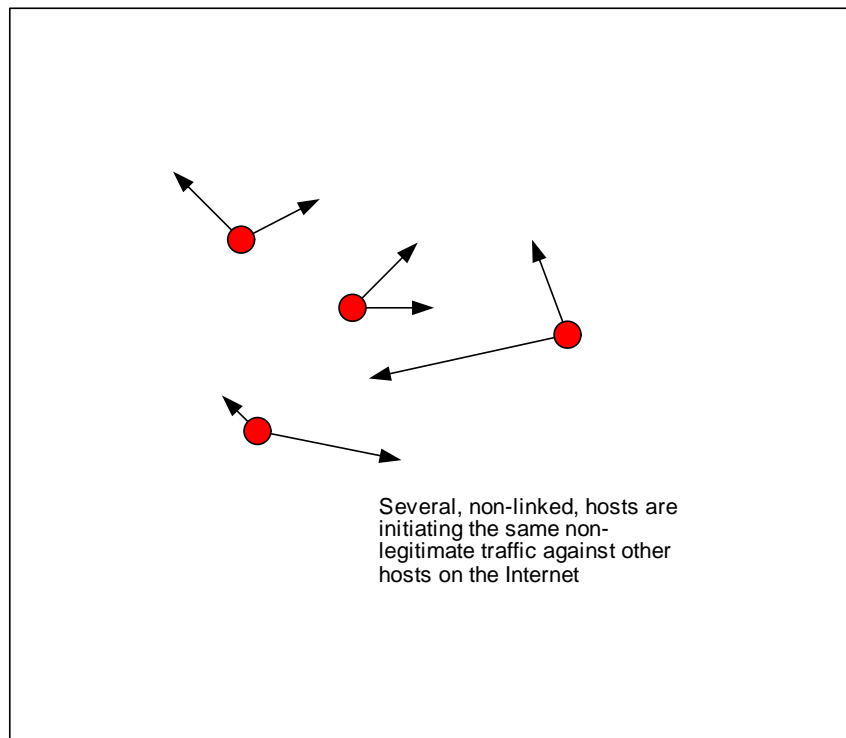


Figure 8

As you can see several, non-linked, hosts are initiating the same non-legitimate traffic against other hosts on the Internet. This might be the case where a number of people that belong to the same social circle / group of some sort start using or testing a new exploit.

Our advantage in this case will be monitoring the initiating systems. Eventually these hosts will go and download the perfected exploit code from some source. By monitoring their activities we can uncover not only the exploit itself, but in some cases the group's main warez server, mail servers used, and nearly everything about the members of that particular group.

Usually the testing stage is regarded as more "safe" than a later stage where the code is used for "production". Therefore the "security" measures taken by the individuals belonging to that group might be less strict.

The last method demonstrated a way not only to trace and track the malicious computer attackers, but also how we might limit the information we need to log and keep performing this kind of operation. The signature matching capabilities, the methods to identify non-legitimate traffic and the ways in which to build this relational database are outside the scope of this document.

There is also a considerable amount of legal and privacy questions that needs to be answered.

Taken together many of the concepts presented in this paper could be used to create a Global *Intruder* Detection System. An *Intruder* Detection System is analogous to the satellite surveillance systems watching troop movements and the building of military facilities in the arena of conventional warfare. It does not look for attacks but tries to pinpoint the activities that precipitate one.

Conclusion

The monitoring of any electronic signaling and communication gives the intelligence community the ability to track and log everything in our “electronic lives”. The information logged can be used to track back malicious computer attackers and exploit code writers. Even if some obstacles are introduced, the smart profiling system (psychology and technology combined), can still discover these malicious entities.

For some people this might look like fiction. Just remember that Jules Verne's books were fiction as well at the time...

Acknowledgment

I would like to thank Saumil Shah, Max Kilger, Lance Spitzner, Jeff Stutzman, Chris Wysopal, Ollie Whitehouse, Colin Gillingham, Phil Huggins, and Brian Carrier for reviewing this paper.