Michael Guidry
April 3, 2017
Defeating digital mass surveillance programs, or the sale of browser history by ISPs within USA and abroad

Deceptive tactics are far easier to apply online than in person. For example, If you attempt to interfere with a microphone in person then it may be possible to hear the background noise. It is far harder to perform the same integrity checks with digital deep packet inspection, or other similar platforms.

I've been brainstorming some ways of deliberately disrupting mass surveillance, and ideas behind digital camouflage regardless of these platforms. The false positives would require far more employees, and processing units to correctly disseminate actual intelligence against fake sessions. It is a way to turn signal intelligence into a battlefield without ever having to send any "real" digital transmissions. The transmissions never reach social media platforms because of the ways they are injected.

Quantum Insert is an offensive National Security Agency (NSA) program which will inject a virus into a web page as it is being downloaded over the Internet. It works by reading a packet which is being sent from a web server to a browser. If it would like to respond to the browser with modified content then it needs particular information from these packets which shouldn't be readily available to anyone except the two parties involved in the connection. It does not work with SSL at the moment due to cryptographic keys not being as easily available. If it can modify the web page then it can force that browser to connect to a third party server. When the connection takes place it gives the ability to infect the browser's machine with a virus. The virus is built with surveillance in mind which makes it of importance while expressing TCP/IP vulnerabilities. These vulnerabilities are significant to the overall mass surveillance scenarios everyone has to deal with in today's day and age. It is purely because the NSA manipulated routers into giving it the information within these TCP/IP headers that allows their systems to hijack these connections to web servers.

The simple packet parsing for the Quantum Insert attack requires parsing the TCP/IP packet's header. It is purely to extract the source IP address, destination IP address, Acknowledge (ACK)/Sequence (SEQ) numbers, source port, and destination port. The ACK/SEQ numbers are the private numbers created within the protocol to mitigate spoofing, and hijacking attacks. Obtaining these numbers is difficult but also the reason why Quantum Insert is possible. It requires either a sniffer either on the web server, or on the browser's machine. The alternative is to have this information from mass amounts of users by manipulation of routers within Internet Service Providers (ISP).

Deep packet inspection which can obtain metadata, and practically all information from e-mails, and other transmissions were difficult until technology's capabilities advanced around 2005. It was tough to find hardware that is specific for performing the tasks required before this timeframe. It is an intensive task for central processing unit (CPU) units on servers, or routers. It is now possible due to companies creating field-programmable gate array (FPGA) alternatives to the initial software implementations. It was infeasible for routers beforehand to simultaneously process gigabits of information while also performing these types of actions. It is becoming easier now, although that doesn't mean that these systems do not still contain the same vulnerabilities.

Operating systems handle packets, and error correction automatically although they only perform these necessary actions for the connections they are personally involved with. An operating system firewall which may use Network Address Translation (NAT) will usually only keep track of packets which may require it to open other ports within the firewall. It is infeasible to process every packet in these ways so they are broken down by services which require it. This is a mechanism which helps the operating systems by only processing necessary packets. It ignores packets which are not being communicated to specific applications on its own system. The internet would be a very slow place otherwise.

It is important to understand why these operating systems don't regularly process every packet completely. It is too much data for the CPU to process while handling its regular duties simultaneously. If you can comprehend this alone then you can understand why it is infeasible to believe that mass surveillance programs can continue to function

properly with the correct mechanisms, or systems in place which wish to exhaust their resources.

Government sponsored surveillance programs may spend a lot more effort, and resources into extending capabilities for particular monitoring nodes. It is especially important if they deem them to be priority, and their signal intelligence from the nodes have been of superior quality. It would continue to be a cat and mouse game if anyone wished to continue to directly attack the systems. It is impossible to find a permanent solution due to the hijacking ability within the TCP/IP protocol itself. This vulnerability will always exist for these remote "invisible" surveillance programs. If these programs were directly on servers, or on client consumer machines then it would be possible to have a secure surveillance system.

The wonderful thing about "attacking" these systems is that you are not directly attacking the surveillance, or government systems. You could perform the attacks on your own systems, or from your own systems just assuming that a surveillance tap is in place. It is not the type of attack that should have any way, shape, or form of expected litigation. You pay for your bandwidth, and it is up to you what you wish to do with it. It is not your fault that their systems for monitoring your data is having problems with the data that you are paying for in every way shape or form to transmit. Most of the programs are run in secret anyways. It is imperative to understand this so that if something does happen to come up later that you recognize that surveillance taps are programs which are meant to be invisible. Its not your fault if their service providers have not created technology capable of dealing with the traffic you wish to transmit.

It is not difficult to inject fake sessions into surveillance programs. You need to create all packets ahead of time, and ensure everything is correct. The ACK/SEQ numbers need to change properly like an operating system throughout every packet. You can send spoofed packets towards either a client, or server depending on the direction of surveillance you are attempting to manipulate. If done properly this will allow exhaustion of resources. You could capture websites which you believe to be under programs such as "Prism," or foreign government programs. It might seem difficult but anyone using the packet capturing

software Wireshark can dump web traffic sessions. If you were to use the Wireshark Hypertext Transfer Protocol (HTTP) filters, then you could modify these packets for manipulation purposes. It would give exact structure of how the packets transit during real sessions. These sessions could be played directly to the wire for injection into these systems. Replaying would have to spoof the destination, and source correctly for each packet. It might be nice to ensure having machines on both sides of the tap. It shouldn't be that difficult, and isn't always necessary. Border Gateway Protocol (BGP) routes have many multi homed scenarios where the packets would come in on the same interface.

It wouldn't take long to create entire platforms across the internet with ability of injecting information onto various surveillance nodes. If you consider the lists of sites under various firewalls such as China, then you can begin to get a concept of which sites are being monitored. It more than likely includes the top social media sites, etc. It requires trial and error as to how many injections, and of what particular information would cause enough false positives. It is also arbitrary to consider the amount of injected sessions required to either exhaust databases, or hard drive space. Hash collisions is an attack that could be used to put these systems to a crawl remotely. Collisions were a factor with some web servers several years ago, and wouldn't be difficult to repurpose for these scenarios. It would take extensive debugging on the surveillance platforms to prepare fixes. It is within the means of a small group to perform these attacks worldwide.

Major effort and resources will be delegated to fixing these attacks. It will revolve around neural networks, and de-duplication techniques. It is quite easy to bypass these especially with a few decent text algorithms. The same circumstances exist with current fake news scenarios on Facebook. If it uses real names, and stories paired with randomization then it will be near impossible to determine. If a neural network is used on the packets as they are being built then it will pretty much score the same on the surveillance systems engines as it is being interpreted. Private corporations, and these government surveillance programs all use the same techniques. The cat and mouse game will always exist.

The real question is would it be illegal for you to blindly send information across routes on the internet with possible keywords such as “bomb?” It would first be difficult to find the origination of these spoofed packets. Spoofing usually doesn’t work without having direct access to the packets of either source, and destination. It does work in this case since you have generated both sides of the connection, and are replaying it to the internet. You aren’t actually communicating with the other server’s software, or even caring if it communicates back. It is a type of blind spoof which only matters to these invisible programs. Would you be held liable for the reactions from programs which may even be illegal? It is quite a conundrum that will have to be explored very shortly in many countries. It would depend on whether or not you are hurting the networks you may be spoofing. If you were to continue to perform it solely on your network then it might not be such an issue legally since you are one of the sessions parties.

It is possible to block these surveillance programs from logging the data being sent across by your computer. You can develop network monitoring, or manipulation routines directly into applications. It could be built into the operating system as well for the entire network packet queue. It can transmit ghost packets using various tactics allowing you to exploit vulnerabilities in the implementations of these programs. It can be done to fill up their queue to exhaust resources. It can also be done single handedly on your own servers, or machines to block their data from being captured correctly. It is not your problem if these programs are not accurately performing TCP/ IP interpretation on every connection. It is quite impossible for them to perform correct interpretation, and verification on every packet for major pipes, or internet providers.

The initial concept which is easy to design would take the packet which is outgoing, and judge by some algorithm when to apply the techniques. For example, it should modify the Acknowledgment number (ACK), and Sequence number(SEQ) so the web server being communicated with would ignore the packet. The surveillance systems however may not verify correctly therefore it would take any garbage data within the modified packet and possibly log it. You may even force that surveillance system to believe the connection has closed. You can then

continue those connections normally. It depends on the implementation of their deep packet inspection but it might just process all packets into the session. It would give a lot of room for misinterpretation especially with protocols such as E-Mail. It is the simplest method which would work on the majority of these systems. It is simple to perform locally, and could be done on every packet. This would more than likely completely destroy the restructuring of these packets for logging or surveillance programs purposes.

In essence, these programs have vulnerabilities which are as fragile as the Internet protocols themselves. It generally cannot be completely fixed, and will always have possibility of manipulation and false information planting available.