GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Security Essentials: Network, Endpoint, and Cloud (Security 401)"
at http://www.giac.org/registration/gsec

Semantic Attacks – What's in a URL?

Richard Siedzik

April 6, 2001

What has been characterized as the third wave of network attacks, are attacks that target people and the way people assign meaning to content. Thus, these "Semantic Attacks" prey on the human element, or more precisely, human nature. It's human nature to want to believe what you see in print. People tend to accept the proposition that it wouldn't be disseminated unless there was some truth to it.

Unlike the first two waves of network attacks the solution to this problem lies somewhere other than "in the math". The first wave of attacks was physical: attacks on the physical components of the Net (routers, switches, servers, databases, etc.). The defenses for these types of attacks proved to be such things as redundancy - removing single points of failure, and distributed protocols - reducing the dependency on any one computer.

The second wave of attacks targets the vulnerabilities of software products, which includes operating systems and protocols. These syntactic attacks look to divert the operating logic of the Net – denial of service vulnerabilities. Defenses for these types of attacks are ever evolving. Operating systems and appliances are continuously being revised and updated to eliminate the potential vulnerability or to throttle back its impact. Thus far, the only real measure of security has been in the detection and response processes.

It has been proven time and time again that the Internet is fertile ground for all sorts of misinformation and hoaxes, capable of costly damage to unsuspecting victims, and unfortunately most times, at little or no cost to the perpetrators. Take the case of the fraudulent press release that caused Emulex Corporation's stock price to plummet 60% back in August 2000. Despite the attacks lack of sophistication, $2.54 billion dollars in market capitalization disappeared, only to reappear hours later. With better planning similar attacks could do much more damage and be much harder to detect.

Semantic attacks on URLs are most common. Spammers have known for years just how easy it is to disguise a URL and just how easy it is to exploit the component which occupies the space between a keyboard and a chair. In the first place, most Internet users have no idea what a URL is supposed to look like, let alone know how to parse one. That brings us to the point of this paper. A little education can go a long way in avoiding these types of vulnerabilities.

So what's in a URL Anyway?

Obviously, the URL scheme in itself poses no security threat. However, it is possible to construct a URL such that it attempts to masquerade itself as a harmless attempt to retrieve a given object when in fact it causes an unexpected and possibly damaging operation to occur.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Although most Internet users associate URL's with WWW addresses, Uniform Resource Locators are more general in scope. URLs are standardized in RFC1738, and in general are written as follows:

<scheme>:<scheme-specific-part>

A URL contains the name of the protocol being used (<scheme>) followed by a colon and then a string (<scheme-specific-part>) whose interpretation depends on the scheme. The best know scheme is the Common Internet Scheme, in which the syntax for the <scheme-specific-part> is as follows:

//<user>:<password>@<host>:<port>/<url-path>

Some of the parts "<user>:<password>@", ":<password>", ":<port>", and "/<url-path>" may be excluded. Only the host part is mandatory. To the untrained eye it's not always easy to interpret a URL. Take the following URLs for example. It's an improbability that anyone who clicks on them would know their intended target.

http://3492563303
http://1066525412
http://2815984929

AT the time of this paper they equated to:

http://www.l0phtcrack.com/
http://www.phrack.com
http://www.sans.org/newlook/home.htm

The first three representations are possible because some operating systems and/or web servers operate on IP address, not only in the form we are use to "aaa.bbb.ccc.ddd" but also on their decimal equivalent. So IP address 167.216.133.33, which resolves to www.sans.org by the way, has a decimal equivalent of 167*256^3+216*256^2+133*256+33 = 2815984929. Not all operating systems and/or web servers will support the IP or decimal equivalent in the URL, especially virtual web servers who require that fully qualified domain names be passed by the browser.

Security Implications

Trust is the underlying security value exploited in semantic attacks. Crafters of malicious URLs exploit the fact that attention is focused on the content frame and not on the location, although they are equally important in a decision of trust. The challenge of avoiding semantic attacks is to recognize when a statement is false, or another way of putting it, to recognize the truth in the statement.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

One of the more recent publicized semantic attacks that took place was the spoof of the Microsoft Knowledge Base articles. Many users clicked on the link posted on a URL starting with www.microsoft.com, never taking the time to examine the complete URL, until surprised by it's content. Full examination of the URL reveled the link to be (no longer active by the way):

www.microsoft.com&item=q209354@www.hwnd.net/pub/mskb/Q209354.asp

The real host of the page was www.hwnd.net. The string "www.microsoft.com" in this case is just a bogus usemame that is ignored by the web server.

To make a URL even more obscure we could craft one such as: www.microsoft.com:bgates@3492563303 or www.microsoft.com@3492563303. Many sites store the HTTP SessionID in the URL, so the above would not appear suspicious. Hence, the URL is easy to read and easy to misunderstand. The URL still points us to www.l0phtcrack.com.

Going even further, with the help of the html language, you can use the anchor tag to display text for a link on a web page that is different than the actual target. With html the syntax in the source page would look like the following:

Richard's Guide to the Great Outdoors

What is displayed on the screen is a link titled "Richard's Guide to the Great Outdoors" but the actual target is http://www.itdn.net. This would not be obvious unless you hover your mouse over the link and observe the actual target, usually at the bottom edge of your browser.

We can explore even deeper the security implications of a URL. Let's take the case of AOL Instant Messenger (AIM) and the vulnerabilities discovered back in December 2000 with AIM versions prior to 4.3.2229. Applications like AIM, when installed, register their URL protocol. On a Windows machine that information can be found in the following registry keys:

HKEY_LOCAL_MACHINE\SOFTWARE\Classes\PROTOCOLS\Handler
and those keys under
HKEY_CLASSES_ROOT\Shell that have a subkey named "URL Protocol."

In the case of AIM its found in HKEY_CLASSES_ROOT\Shell. AIM registers the URL protocol "aim:" as a hook into its executable. This allows AIM users to publish their screen name on a web server and be added to a viewer's "Buddy List".

In order for this to work each "aim://" URL gets passed directly to the aim client, as if it were typed on the command line. Hence, typing "aim:goim?Screenname=rich&Message=hirich" in Internet Explorer will pop-up an instant message box ready to send to rich.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

AIM client software prior to version 4.3.2229 contains numerous vulnerabilities that allow a maliciously crafted URL to overflow internal buffers and obtain control of the program. Her are two examples of buffer overflows demonstrated by typing the following URL into your browser:

> aim:goim?=<insert 300+ string of AAAAA here>+-restart
>
> aim:buddyicon?screenname=abob&groupname=asdf&Src=http://localhostAAA … where there are > 300 "A" characters.

It is important to note that you did not need to be running AIM but merely have it installed to be vulnerable to these attacks. It has been demonstrated that by receiving malicious email or visiting a malicious web sites the vulnerabilities of AIM enabled the execution of arbitrary code on a target machine. Needless to say, it may be important to know what other protocols might be registered on a particular machine, especially when you presume to secure it. As in the case of AIM, most firewall configurations do not guard environments where the communication is first established from inside the firewall.

And if all of this wasn't enough, it's possible to include encoded characters in a URL. URL encoding is required for many special characters but can be applied to regular alphanumeric characters as well. How many Internet users will be able to parse a URL whose <scheme-specific-part>shows up as hex codes?

Defenses

False statements will continue to be inserted into the network as real ones. Vulnerabilities will probably continue to evade our technological defense mechanisms for sometime to come. So how then do we adequately protect our networks today from these types of attacks?

Certainly, an ideal defense system is one that is wired into the network; one that senses various matches and/or mismatches in program code and/or content and takes appropriate measures to isolate the contaminate. But how do you effectively accomplish this when the target of an attack is not a technological component of the Net?

Content filtering products that screen information coming into a network do a fairly good job in detecting keywords and phrases in email messages and attachments. Such products filter emails for viruses, Trojans, spam, etc. The draw back to these types of products is that they are not very proactive. An administrator must set up rules to detect and block messages by these keywords. Another drawback is that the rule set sometimes blocks email messages that should be allowed through because they just happen to contain a word or phrase that matches the rule.

Intrusion detection system's both network based and host based do very little in the way of offering any protection from semantic attacks. Network intrusion detection systems

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

monitor network activity looking for signs of network attacks, network misuse or anomalies. Host based systems monitor traffic into and out of the host. Some can even be configured to audit key system files to check for signs of tampering but very little in the way of verifying the authenticity and accuracy of a URL that ultimately shows up in an email message on someone's desktop.

There are some other products, although not originally intended to serve the specific purpose of defending the network from malicious content that can use to support the effort. Specifically, on our network we have a bandwidth-managing device installed to shape the traffic going over our wide area links. With this device we have the ability to screen, redirect, and even block incoming and outgoing connections based on protocol or URLs. On one occasion we used it to screen incoming URLs to check the validity of the "aim://" URL and guard off the AIM vulnerability described earlier in this paper. However, devices such as these should only be called upon as a temporary measure. A device such as a bandwidth manager was designed to manage bandwidth between networks and not to provide security for them.

Today's Content Filtering products and Intrusion Detection systems are certainly part of the solution. However, they are just not up to the task of adequately protecting against semantic attacks. They do, however, provide a layer of protection that organizations cannot afford to be without. As time goes by you will see these products become "New and Improved" but still you won't see them become human centric.

To augment whatever software or hardware defense systems we choose to put in place, we need to initiate a human counterpart, one that educates and transfers knowledge of such things to the overall user community. By this I mean more than just issuing guidelines and policies. When was the last time you got a call from a user asking you to clarify something they read in the latest policy update? Do you really believe the general population understands these things without questions?

It takes some human intervention by the IT or network security staff in the form of face-to-face information sharing and training to bring the average Internet user up to a level where they can protect themselves from themselves. Everyday, Internet users, consciously or unconsciously, make security decisions every time they decide to trust what they see on a screen and click on a URL. It's ultimately up to us (the IT community) to see to it that they have been given fair knowledge in order to mount an adequate defense.

In many organizations the Internet connection is primarily and end users tool as apposed to a business-to business medium. It becomes an IT role to train individual users to be more responsible users of the Internet. Instruction as apposed to documentation is the key. Obviously, with no commercial solution available to adequately protect our networks from semantic attacks, we have no choice but to become the engineers and design and implement an Instructional defense.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Many of us in the IT field believe that as long as we, the caretakers of the "information utilities", are informed and trained the more secure our networks become. Well that's probably only half right. The reality is …the better the users of our networks are trained and informed the safer our networks become.

References:

Bruce Schneier, Crypto-Gram, Feb 2001, "A Semantic Attack on URLs"
http://www.counterpane.com/crypto-gram-0102.html#7

URL, URL, Little Do We Know Thee, By Razvan Peteanu
http://securityportal.com/articles/urlurl20010307.html

RFC1738
http://www.securityportal.com/rfc/rfc1738.txt

PFIR Statement on Internet hoaxes:
http://www.pfir.org/statements/hoaxes

@stake Inc., www.atstake.com, Security Advisory
http://www.atstake.com/research/advisories/2000/a121200-1.txt

MSDN, "Registering an Application to a URL Protocol"
http://msdn.microsoft.com/workshop/networking/plugga
ble/overview/appendix_a.asp

Semantic Attacks: The Third Wave of Network Attacks
http://www.counterpane.com/crypto-gram-0010.html#1

The conceptualization of physical, syntactic, and semantic attacks is from an essay by Martin Libicki on the future of warfare.
http://www.ndu.edu/ndu/inss/macnair/mcnair28/m028cont.html

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.