

**Emerging Cyber-Threats on the Web
And what Google is Doing to Help**

Dr. Neil Daswani

15 September 2008

Voice: -- on the part of the Air Force Association Staff, and I'd like to welcome you all to our Air and Space Conference and specifically here to this talk today by our Google - I've been calling him the Google Man all day - but it's Dr. Neil Daswani. I won't read his bio to you. It's in the booklet so you can read that. But we're very excited to have him talk with us, kind of a corporate industry perspective on cyber security and how these things are relating.

Also I have a book here, you might want to write this down. It's brand new, *Foundations on Security - What every Programmer Needs to Know*. Brand new, just fresh off the press. Dr. Daswani has written it, so you might want to put that down and Google it - ha ha. [Laughter]. Find it on Amazon.

Dr. Daswani: Thank you, Lois.

Good afternoon everyone.

First of all, I just want to thank the Air Force Association for the kind invitation to speak today. I do hope to give you a corporate view on security, tell you a little bit about the things that we're dealing with in the private sector, and I will do my best to highlight relevance to the Air Force as well as the military and our government organizations whenever I can.

I will be talking about a variety of things. I'll be talking about some cyber threats. I will also chat a little bit about what Google is doing to help. Google is doing actually quite a few things in the security space. I won't be able to talk about all the things that Google is doing to help, but I'm hoping I'll be able to chat about one or two of them. Then of course I'm hoping to have a good maybe 15 to 20 minutes for Q&A afterwards, so if you have other questions about what Google is doing I'm more than happy to try to answer as many questions as I can.

At Google I am a product security manager. What that means is that Google has a number of products. You've probably seen Google Search, but Google of course offers e-mail; Google allows you to use Google aps, edit documents,

do things like create presentations on-line. It has a whole variety of products and we need to make sure that each of them are secure.

The bad guys look at Google as a big target and they want to try to abuse our services in various ways. So I together with our security team work with a number of the product groups at Google to secure our various applications.

I throughout the talk am going to mention a number of different resources. I may point to some statistics. I might point to various reports. I might point to other slide presentations and what not. All those can be accessed on my web site at NeilDaswani.com. There's also a blog off of that site at NeilDaswani.Blogspot.com, so you can feel to check those for additional resources.

During the talk I'm happy to answer questions as well, especially if you have clarifications to ask. I'd definitely like to do some Q&A at the end as well.

What I'm going to do is chat a little bit about data breaches that have occurred over the past couple of years. I'm then going to go into talking about hacking, one of the major causes of data breaches, and I'm going to talk about web application vulnerabilities. What that means is when the web was first constructed, the web was constructed to serve documents, right? When the web was initially designed by Tim Berners-Lee & Company. Over time the web evolved into a platform which allowed people to do transactions, commerce transactions as well as other types of transactions. What's happened is that web sites have no longer become repositories of documents, but they've become full-blown applications, just as if you were to launch software applications on your own client PC. With that transition comes a whole bunch of vulnerabilities. So I'll chat about some of the web application vulnerabilities that plague some of the applications on the web today. I'll chat a little bit about what software professionals can do to help.

In fact, before I proceed what I wanted to ask is how many of you would consider yourselves actually non-technical? Okay. And how many of you have ever written a piece of software in the past. Good. How many of you would consider yourselves technical, meaning either you've written a piece of software in the past or you're involved in some other technical field? Good. It looks like I have a mix of about half/half. I'll do my best to try to keep things as understandable to the largest percentage of folks as I can.

So I'll chat about what software professionals and technical people can do to address that half of the audience.

After talking about that I'm going to go into malware distribution. Malware, of course, is malicious software, and it's been with us for a long time. But the way that it's transformed and distributed these days is very different than the ways that it has in the past. So I'll chat more about that.

Secretary Donley had mentioned what are some of the big frontiers we need to worry about. I believe this is one of them. I'll tell you more about that and why in just a bit. And we'll talk about what Google is doing on that front.

Finally, for the benefit of everybody as end users I'll talk about some things that you can do to protect yourself as you're using the web and other systems. Of course I know when you're using your government military systems there's a whole other set of rules, so I'll describe things that I think will be applicable, things where you're web-browsing at home, for instance.

Let me kind of get underway.

As somebody who's been in the security space for some time, I picked up my PhD from Stanford and did my dissertation on security about four years ago and was even involved in developing security software even prior to that at places like Bell Communications Research and Lucent. After looking at what's been going on the past three or four years, I kind of wonder whether or not the sky is falling. So let me talk a little bit about why I have this concern.

Let me just walk through some of the most spectacular recent data breaches and tell you a little bit about how they happened.

I'm going to start by talking about a company called TJX. It's a holding company for a number of retail department stores - TJ Max, Marshall's. How many of you have ever shopped at any of these stores? Okay. All of your credit card numbers have been hacked. If you don't have identity monitoring services, you should get them.

The important thing to talk about is how did that happen? Over 45 million credit card numbers were stolen from this company. That's a lot of numbers. How does

something like that happen? How do you lose and have 45 million credit card numbers stolen from you?

What happened is that the retail department stores were using wireless technology, they were using WiFi to transmit credit card numbers from the point of sale where you swipe your card, to the back end server so that they wouldn't have to deploy cables at their stores. They were attempting to protect these communications using a protocol called WEP, the equivalency protocol that was actually built into the 802.11 standard. Unfortunately, the technical community knows that this protocol was vulnerable to certain attacks as early as 2002. Nevertheless, TJ Max and company continued to use these protocols to attempt to protect the traffic.

The problem is that due to some of the vulnerabilities in this protocol, what the bad guys could do is drive their cars outside the stores and/or park them in parking lots. They can have laptops in their cars, and all they need to do is record the wireless data traffic that's going by. Once they've gathered enough traffic what they can do is reverse engineer the encryption keys that are used. So all that you need is typically a couple of hours worth of traffic, recorded on your laptop, and use some common, off the shelf tools to figure out what the key is, then you've basically got access to the data.

There was an organized group of cyber criminals that did this for stores across the country and aggregated all this data together. They had a pretty sophisticated network where some of the folks would drive the cars, gather the data; others would use tools to decrypt the data; others would, once they got the credit card numbers, burn them onto blank mag stripe cards; others would take those mag stripe cards and enter them into ATMs and get cash back. There was a pretty significant undertaking to get this done.

One of the trends here is that the bad guys typically used to be teenagers that wanted to go out and create mischief. The big trend here is that they're now organized cyber criminals that are conducting cyber attacks for money, for financial gain.

In this particular case there were about 11 folks at the heart of it and they've been convicted, but it took about a year to find them and convict them.

What's the solution here? Well, don't use WEP, right? You need to be very careful about what protocols you're using when you're using wireless technology simply because

data gets transmitted over the air and anybody within the vicinity can grab that traffic.

That's TJX. That's one case. Let me talk about a couple of other cases.

The year prior to that the Department of Veterans Affairs, the VA, would enlist the services of a number of different subcontractors. In one case the enlisted Unisys, a very well known computer company, to conduct some services for them. There was an employee at Unisys that had a hard drive and/or associated media that had very personally identifiable information about veterans stored on it. And because of the fact that we can store so much data in such small form factors these days, he took a hard drive home that had data for 26.5 million veterans on it, and he wasn't authorized to take this information home, but thought it would be convenient, or did it for whatever reason, and it just happened that one of the days when he took this data home, his home got burglarized. Somebody broke in. They stole the hard drive in the laptop. The burglar may or may not have known that all the valuable data was on it, but nevertheless, once this occurs, due to the fact that there are data breach laws deployed in various states, such organizations are required to notify consumers when it happens.

What was the data that was stolen? The names, dates of birth, social security numbers, address, and insurance information for 26.5 million veterans. I don't think this is the best way to of course treat our veterans, to have their data unencrypted and outside of some kind of confines.

What happened here, this was extremely serious because of the fact that this is exactly the kind of information that you need when you say go apply for a loan.

In any case, the employee was dismissed; the employee's supervisor resigned; and this caused quite a bit of a stir.

Given the fact that this happened back in 2006, I believe that Congress actually started working on a federal version of the data breach laws which effectively bring the same kind of requirements to a larger community of states. Of course the problem is I believe that that legislation is a bit more diluted. This is what happens in committees.

So this is another example of something that's gone wrong.

A third example of something that went wrong the year prior to that was the Card Systems data breach. How many of you have heard of Card Systems? No one. Very few people. Actually, to tell you the truth, even those folks that have heard of Card Systems probably wouldn't have heard about it if a particular data breach didn't occur.

So Card Systems was a credit card payment processing company. When you swipe your credit card at a point of sale there's a whole bunch of back end servers that that credit card number goes through to get authenticated. And Card Systems ran one of the gateways on that path to authentication. They had a database of about 43 million unencrypted credit card numbers. That wasn't a problem in itself until somebody at the company decided to connect a web site to that database. So they would allow people to register on the site and maybe see their transaction histories and enter their user names and passwords to log in and such things. Of course this wasn't a problem until the bad guys went to those web pages and instead of entering user names and passwords and such things, they entered database commands and were able to get those database commands to successfully execute on the back end systems at Card Systems.

What they did is they dropped a script on the back end database which said please e-mail us a couple thousand credit card numbers once every day. So the database went ahead and did that. This went on actually for about six months before Card Systems even noticed. Once that did happen, of course Visa and MasterCard canceled their contracts and the company's revenues went to zero overnight. But I think the real victims here are end consumers, right? There were a number of credentials that were actually stolen, but because all 43 million credit card numbers were in the database and it was kind of hard to tell which of the 263,000 had been stolen, all customers, all 43 million had to be notified. Most of those numbers had to be changed, et cetera. So this creates a lot of pain.

I hope you can see why I look at this, just over the past couple of years, and think the sky is falling.

The message, though, that I would like us to take home is that we don't have to let the sky fall. We can do a lot together as a community to prevent the sky from falling, and I'll chat about some of those things in just a bit.

I gave a couple of examples of data briefings. You might say well Neil, sure, these are just three cases.

They happen to be big and spectacular, but is this problem really that bad?

So I had one of my interns pull some statistics from a database at PrivacyRights.Org, where they've been keeping a very good chronology of data breaches that have occurred over time, since 2005. Since that time, over 230 million customer records have either been lost or stolen.

One of the fields that the list in this database is why was it stolen, how did it get stolen? And so from this pie chart you can see that a majority of those credentials were stolen by a hacking, and I'll talk about what I mean by hacking. There's actually many different ways to hack and steal information. I'll talk about just a couple of them. The remaining reasons are due to stolen equipment and lost equipment.

So IT organizations can do more to prevent equipment from getting stolen, or even if equipment does get stolen it's important to make sure that any data that's on that equipment is encrypted so that even if the bad guys get the data it should be useless to them. This of course, for those of you that are in the technical community, know that you shouldn't keep the encryption keys on the equipment as well. The key should be somewhere else. Otherwise the bad guy can just use the key to decrypt the data and that wouldn't help.

With that said, let me talk a little bit more about hacking.

What do we mean by hacking? I had mentioned the Card Systems case. Let me give an example of how a particular hack can happen.

You can imagine that there might be a web site which allows you to log in by say entering your user name and your password. Most people will go ahead and actually enter their user name and password. Of course the attackers don't do that. What the attackers do is the attackers see this as a window into attacking your site. In this particular example an attacker has entered a database command inside the user name field. Now you might ask okay, great, I could enter a database command, but why the heck would the web site and/or the database actually execute that command? Well, the reason that this is dangerous is because the bad guy in this case has put in a little bit of glue here so that if the web site was not written correctly, if the programmers were not aware of certain kinds of attacks, and the bad guys specify a little

bit of glue to try to get this database command to execute, they can be very successful in doing it.

So the idea here is that the bad guy has specified a database command as well as some data just before it and just after it so that the database command can get executed successfully.

The attacker can enter something for the password. It doesn't really even matter what. The second the attacker hits that login button some bad stuff will happen. Let me explain exactly how this takes place.

I'm going to use some diagrams to show how certain types of attacks happen. These diagrams are going to involve a number of different parties, so you might have a user that's using a web browser, you might have the web server, you might have the database, and the way to read these diagrams is it just goes, in terms of time, from the top to the bottom. The event that happens up here happens first, and then other events happen later.

So in this particular example it just shows what is the regular login process when you log in to a web site. The way it works is you as a user will enter a user name and password. The web server may take that user name and password and the web server needs to figure out if that user name and password is authentic. Is that a legitimate user name and password?

In order to answer that question what the web server does is it constructs a database command, namely "select password from users where the user name is what the user typed in". The whole idea here is that it does a database lookup to see if there is a password on file for the user that was specified. The key thing to take note here is that this [dollars] user name is basically a variable that gets filled in with whatever the user types in.

So in the case that a good user, let's say Alice decides to log in. The web server will ask the database, hey, can you look up the password for Alice and then it can match it and see if it's the correct password. If the user name and password checks out then it returns the user a web page, say their bank balances or whatever it happens to be.

What happens when the attacker enters this? Well, this input from the attacker gets stuck in for this variable and what gets constructed is a database command where it tries to look up a user, but this quote and semi-colon kind of ended or completed a first database command. Then the rest of the attacker's input is basically a second

database command. This database command says "drop table users". That's an instruction to the database saying please delete all information about all users in the database, right? Well, because of the fact that this input was not checked by the programmer, it was just kind of fed directly to the database, this will actually execute and it will eliminate all information about all users from the database. This is the type of an attack called a "denial of service" attack. Any legitimate user that attempts to log into this database afterwards will not be able to because their information has been erased.

Now that might seem bad in itself, but you might argue oh well, you can just restore the data from backup or what not. But the problem is, that legitimate users have been impacted. An attacker should not be able to do it this easily.

So with various web sites you need to be very careful about how they're programmed. The second you take a web site and make it available to the world, you don't know who's going to be using it and you have to make the assumption that you cannot trust them, and that they are not going to do nice things. So this is an example of an attack called a "SQL injection attack". SQL stands for Structured Query Language. It's a database language. But the idea is that the bad guys are able to inject their own commands into your database, so this is an example of one way that attackers conduct hacking.

These types of attacks have become so popular that there's even cartoons about them in the technical community. So in this particular example, mother is receiving a phone call from her son's school. She asks oh, what's wrong? Did my son break something? The school says, kind of, in a way, yeah. And asks the mother, did you really name your son "Robert";droptablestudents? Basically the son was attempting to hack into the school's database and eliminate the information. So she says, oh yes, we call him little Bobby Tables. So of course the administrators are upset. But she says you know, you should protect your systems a little bit better, right? That's a thing to keep in mind.

That's one example of how hacking can happen. Let me give one more example. By the way, I'm giving examples for e-commerce type systems, but keep in mind all these things are applicable to many other types of systems as well.

So in this second example which is called "cross site request forgery" in the technical community, let me kind of

explain what it is in English to the best of my ability here.

Assume that a good user, Alice, is using a web application at bank.com. She basically logs into her bank. She goes to bank.com to check her balance. The way that most web systems work is that when you go to the front page you enter your user name and password, right? The web site gets that user name and password and checks it and if indeed you're an authentic user, what the web site does is the web site gives your web browser a cookie. How many of you ever heard of the term cookie? Good, everyone's heard of cookies. So this is one of the things that cookies are used for on the web. When you log into a web site you provide the user name and password, the web site gives you back a cookie. The reason it gives you back the cookie is so that when you go to the next page and the next page after that, you don't want to type your password again. So the idea is that your web browser, instead of sending back your user name and password each time, sends back this cookie so that you don't have to log in to see each page.

Remember, the web page was based on a protocol that was used to serve documents. It was not built initially to do commerce transactions. So these cookies allow you to do commerce transactions by automatically supplying some data back to the web site when you interact with it.

So let's say that Alice logs into her bank and she gets back this cookie. Let's say that in some other browser window she gets lured to some malicious web site. Let's say she gets lured to evil.com. So how does she get lured to evil.com? There are many different ways, right? She could be reading her e-mail in another window and click on a link in a fishing e-mail or something like this and get lured to evil.com. She could also end up going to some other web page that's infected, and I'll talk about what that means. But there's lots of ways to lure users to malicious web sites.

So the idea here is she's logged into bank.com, and while she's logged into bank.com she gets lured to an evil site. Let's see what can happen.

As I mentioned before, Alice goes to bank.com and says I would like to log in. The bank gives her a web page that has a user name and password field on it and she fills in her user name and password and supplies that to bank.com. Basically her browser calls some authentication script on the bank's web site and says here's Alice's user name and password. Alice loves Bob and that's what she uses for her password.

Bank.com looks up her user name and password in their database and says oh, yes, Alice is a legitimate user of our web site, and so I will give her back this cookie that her browser can use on subsequent interactions with my web site so that she can check her balance and do other such things without having to type her password in on each page. So she gets back a web page.

She may then, at some subsequent point decide to check her bank balance. So her browser, she clicks on a link that says "view balance". Her browser says okay, call the "view balance" script at bank.com, and because Alice is already logged into bank.com, her browser automatically sends over the cookie, right? As per how the web is supposed to work.

Bank.com says oh, I recognize this cookie. I've seen this. I just gave this to Alice. It must be Alice who's trying to check her bank balance. So the web site will look up the bank balance and tell Alice, hey, your balance is \$25,000. So this is how things work in the normal, nice, hunky dory case.

What happens now if in a second browser window she gets lured to a site called evil.com? Well, let's look at what can happen.

As before, she logs into bank.com. She is provided with a page that allows her to enter her user name and password. She enters the user name and password. Her browser calls this authentication script at bank.com. She gets logged in, she gets a cookie, so that on subsequent interactions she can just see her balance or whatever it happens to be. But let's say at this point instead she gets lured to evil.com.

What can happen is she might request some web page from evil.com. It might be called evil.html, and in this web page from evil.com there comes some - html - how many people know html or have heard of html? Most people. Html is the language of the web. It's the language in which web pages are written.

The way the web works is that when you look at any typical web document, one of the great things about the web is in constructing one web page you can fetch an image from one part of the web; you can fetch text from another part of the web; and you can kind of see all this stuff on one page.

So what's happened in this case is that the web page that evil.com has returned to Alice includes an "image" a specified by this html code. Alice's browser wants to faithfully render this web page for Alice. So the browser says oh, okay, there's an image on this page. I should go ahead and request this image from wherever it is on the web. In this case the attacker has said well, there's an image at bank.com that I want you to go get. Of course this really isn't an image, but the browser doesn't care. It's going to go ahead and try to fetch whatever resource, whether it be an image or a piece of text or whatever, from that page.

So let's look at what happens when the browser basically makes a request to bank.com.

What's happened in this case is that the attacker has specified this image which tells the browser, "please go fetch this image", but it really isn't an image. It's really a request to bank.com's pay bill program and the attacker has specified some data along with this request. The data basically says where the bill payment should go to. In this case her browser wants to render this web page, so it tries to fetch an image from bank.com by calling the pay bill program, and it specifies the attacker's address.

Now because Alice is already logged in to bank.com, Alice's browser includes the cookie with the request. If you remember what the cookie is used to do is it's used to authenticate Alice. So when bank.com gets this request it looks completely legitimate because as long as that cookie is there it's Alice as far as the bank is concerned. So the bank will go ahead and serve the web page back, and in the process will go ahead and issue this bill payment.

The web page that Alice's browser gets back from evil.com might be broken but it really doesn't matter. The damage is already done. A payment of \$10,000 went to 123 Evil Street.

The way this plays out on the internet is that what the bad guys do is, if you're a bad guy you won't really do it in this way, right? You don't want the check coming directly to you. Alice is eventually going to complain to the bank and the bank is going to investigate and try to figure out where that \$10,000 went. But the way the attackers really do this, because they don't want to get caught, is they will post web pages for fake companies and they will put a job description saying well, you need to know how to use Microsoft Word. And you need to have an

internet bank account. And you need to be able to receive checks.

Basically what the cyber criminals will do, is they'll effectively hire mules. They'll put up fake web pages with these job descriptions, people will apply, they'll arbitrarily select someone. The person that signs up for this job thinks oh, I have a nice work from home job. I receive checks in the mail, I cash it, I keep ten percent for myself and I give the 90 percent to wherever I'm instructed to give it to. So this is how these attacks kind of get completed.

This is big business. It seems silly, but it's really happening out there.

The question is, what can we do?

If you're either a software professional, or let's say you're responsible for acquiring systems, you should arm and educate yourself. So the way to think about this is that most engineers, most computer science, most folks that go to school for computer science are typically not required to take a security course to get their degrees. As a result what happens is they graduate, they join very reputable companies, but they either don't know about security or learned about it in kind of an ad hoc fashion, and when they write their programs that run on web sites they're not aware of all the threats and all the ways that the bad guys try to abuse them.

So it's extremely important that over time we, in our universities, both the military universities as well as our other university institutions. Over time we require these types of courses.

That said there are about 15 million programmers in the world right now, generating over a billion lines of code a year. Many of them are for web applications and were not aware how software can be abused. So it's important to educate oneself.

I've included a couple of URLs where folks can learn more. If you're a technical person, these are great resources for you to check out.

This site, LearnSecure.com has basically a bunch of links to places where you can go on the web to learn more about security. On Google, on code.google.com.edu we have made a web security course available for free. There are over 500 slides available so if you're in a position where you teach or you need to influence others to teach

security, this is a place where you can go. Actually, these slides correspond to each of the chapters of my book.

When I was working on my PhD at Stanford I had developed these courses and Stanford uses them to make available instruction, but I basically have made all these materials available for free. Of course if you are interested in a place where you can learn more, at Stanford at this URL we have an advanced security certification program that we've developed that covers things like emerging threats and defenses, that teaches people how to build secure web applications, that tells you how to write secure code. I have some brochures for that particular program if you're interested. I've left them by the side.

The first step is education. In fact on every either software project, if you're developing code it's important to say elect a security czar that keeps tabs of these things. If, on the other hand, you're in the business of acquiring technology then it's important that when you go to your vendors, you need to ask them about these things. You need to vet the software and make sure it can defend against these kinds of application security threats.

Most of the time when you go to a vendor and they ask you about security they'll tell you oh, we use this firewall. All of the attacks that I just showed you went right through the firewall. They were all conducted over the web. And what do network layer firewalls do? They let web traffic through because everyone has to conduct business. You can't block web traffic. So the attackers have basically been channeling their attacks over web traffic. In order to defend against those kinds of attacks we basically need new types of defenses. So if you're a software professional, these are examples of some of the things you can do.

If you're a manager, either in your building software or again you're responsible for vetting vendors, it's important to make sure that your vendor has a development process that's instrumented for security. I'll talk about what that means in just a second.

It's also important to organize for security, make sure that you're using any security professionals that you have in-house appropriately, and invest in training. Secretary Donley said it's important to invest in cyber education. I believe that is right on the mark.

Let me give an example of what I mean by instrumenting a software development process for security.

How many of you are familiar with either the waterfall model or incremental software development models, ad val models, these sorts of things? A couple of folks in the room. Basically the idea is that when you develop a piece of software, just as with any other engineering task, there are a number of steps. You want to probably gather requirements about what you want to build, you want to do some architecture and design, you want to write up some test plans so that you can check that whatever you thought you were designing actually got built. You want to of course write the code. You want to then run your tests on the code and you want to gather feedback from the field. The key point is that at each of these steps there should be parts of these steps that are focused on security.

For instance, when you're developing requirements, or when you're developing an RFP about a system that you need to buy, you need to put in not only the functional requirements, what do you need the system to do, but you also need to ask about the security requirements. What software developers, for instance, need to do is they need to think about abuse cases. Not just how people will use the software, but how people will try to abuse the software. That's just an example of something you need to do to instrument your software development life cycle for security.

I'm going to shift gears a bit and I'm going to talk about how all of the previous threats that I mentioned have resulted in some fundamental changes in what the bad guys are doing with regards to distributing malware.

How many of you have ever had a virus or a worm on your computer? That's a pretty high number. As if it wasn't bad enough, the bad guys are getting smarter with regards to how they deploy their malware.

What does malware do? Let me give you an example of what are some of the things that malware is doing these days.

When malware infects your computer, it typically used to spread to other computers. That was at a time about three or four years ago when the malware was written by teenagers who just wanted to create a name and some fame for themselves.

What's the big change in malware distribution? Malware used to get distributed either via say e-mail, you'd receive an e-mail attachment and if you double clicked it and opened that file, that virus would read whoever's in your address book and it would spread to other

folks by e-mail. That's one way it would spread. Or it would infect your computer and then look for other computers to infect on your local area network.

Those methods of spreading are what used to happen yesterday. One of the big shifts that's occurred is that malware now spreads using the web as its primary platform.

What the bad guys are doing is that they will compromise a web site that say potentially gets a good amount of traffic, and instead of defacing the page and saying hey, I broke into this web site like the teenagers used to, what they do now is they just very silently and quietly add a little piece of code on the web page such that when your browser visits that web page, just visits that web page, no user interaction required, that web page will cause your computer to contact some server in China and download a malware binary and install it on your computer. So you just visited a web page, that's all you did, and your computer gets taken. This is the way that malware is spreading primarily these days.

I could of course go into the technology of how the bad guys are doing all of this, but I think I'll spread you those details. This is the reality.

In fact let me talk a little bit more about how the bad guys are scaling their attacks. It used to be the case that maybe a bad guy breaks into one site at a time and infects web pages such that users who visit those web pages get infected. And you can imagine that the bad guys will want to break into a couple of big sites that get a lot of traffic so that they can infect many users at a time.

What they started doing is they've started weaponizing and automating their attacks so that what they do is they don't break into one web site at a time, but they'll break into hundreds, thousands, tens of thousands, hundreds of thousands of web sites in one shot. Let me tell you how you do that.

What an attacker will do is an attacker will make queries to a search engine like a Google or like a Yahoo, and the query, the search query they will enter will be a search query that will ask the search engine, hey, please tell me all the web sites that are say using Microsoft SQL server, Microsoft's database and/or some additional software that indicates that it might be vulnerable to say a SQL injection attack, kind of the attack that I told you about earlier which was used for data theft. Except instead of the data theft, they take advantage of the fact that most web sites, when they render web pages for you,

those web pages are constructed based on data from a database. Most of that data isn't coming from individual files any more. It's from data in a database.

So the attacker will query search engines for vulnerable sites and they will, after their programs, they basically construct programs that do this. They don't do this manually. After their programs query for the list of vulnerable sites, the programs will then automatically attack those sites and inject some additional malicious html code onto those web pages.

So what the attacker has here is a program that can find lists of tens of thousands of vulnerable sites, and then inject some data into those databases such that when people visit those sites their computers will get infected by malware.

So what the bad guys have done is they have taken an attack that typically you'd only want to do on large web sites, and now they'll do these kinds of attacks on many many many small web sites. So I don't care that a web site doesn't have a lot of traffic any more, as long as it has some traffic this is fine because I can compromise large numbers of web sites all in one shot.

So what happens is once these sites get infected, when users simply visit those sites, they view pages on the sites, and they get infected. Once they're infected, their computers join a bot-net and the compromised machines just ask the attacker's server, hey, what should I do today? Do I log keystrokes, do you want me to do a denial of service attack, as happened in Estonia, et cetera? So this is pretty serious.

In fact in the first half of this year, 2008, there were over three million web sites that were infected in this fashion. That's pretty significant.

Let me chat a little bit about what Google is doing to help. I think these kinds of things can really threaten our commerce infrastructure. So what is Google doing to help?

What Google is doing to help is that when you say enter certain search queries, you might get back some results. I've actually intentionally left off the search query that was used to get this particular result, but the idea here is that if Google detects that one of the results may in fact lead to a web page that can infect your computer, what Google will do is it will add this note, this annotation to the search result saying "This site may

harm your computer." The hope is that somebody won't click on this result link, right, if they see that the site may harm their computer. But some people really want to get to these sites so they click on it anyway, at which point Google pops up a page that says really, we're not kidding here. [Laughter]. It's funny.

When Google first started doing this kind of black listing of web sites it would make this link linkable, right. People would still go ahead and click through, so Google said finally, okay, look, we're not going to make this linkable. If you really want to go there you're going to have to copy and paste this into your address bar. My hope is that people are not doing that. But again, there's no way to tell.

There's an organization called StopAdware.org which has been working with Google, and Google has been providing data to StopAdware.org to help webmasters that have their sites compromised recover from these types of issues.

So this is an example of what Google is doing to help deal with malware threats.

For those of you that are technical in the audience you might be interested in how the heck does Google do that, right? The way Google does this is Google has an index, a repository of billions of pages on the internet. Google has a set of technology, a set of what are called "map reduce jobs" that go through the billions of web pages on the internet to find which ones of them might be suspicious. For instance, does the web page have some obfuscated or malicious looking html and/or java script? It basically pares down from the billions of pages in Google's index, to figure out what are say the tens of millions of sites that might be suspicious. It then brings up those sites in virtual machines running on Google's server farm to identify if say a new binary comes down to the virtual machine just by viewing the web page, and then basically creates a malicious age repository so that every time a search happens in addition to giving you search results, the results will look up in this malicious page repository, all within just a couple of milliseconds, and the results are given to you with annotations saying this site may harm your computer.

That's an example of what Google's doing. Let me tell you a little bit about some future directions that the attackers are going in.

The attackers always used to do phishing attacks. How many of you know what a phishing attack is? Most of you.

You get an e-mail, it has a link to your bank web site, except it's really not your bank's web site, it's an imposter web site that say encourages you to type in your bank's user name and password.

What the bad guys are doing is they're getting very smart about their phishing attacks, and they are sending out e-mails which have say breaking news alerts with a whole bunch of headlines that are constructed to encourage you to click on things within the e-mail. For instance, some of the headlines that they use, and they of course test these with actual users or with compromised users and do a run-off, a bake-off. But you can imagine, "how to save money on gas" is going to be very interesting to many people. Our putting in things like "McCain gives up fighting for the presidency." That might seem interesting to a lot of people. They'll just click that.

So the idea is they'll use this in the subject line and they'll make it look like the e-mail came from MSNBC.com and if you look at this e-mail, this is a phishing email except it's not your traditional phishing e-mail. It doesn't try to target you and figure out kind of what bank web site you go to, but instead there will be a couple of links in the e-mail. Some of the links will actually be legitimate. This link here at the bottom will really take you to an unsubscription page at MSNBC. This link here at the bottom will really take you to Microsoft's privacy statement. But this link up here, which is the link that everyone is most likely to click on because it's right below the breaking news alert, will take you to a web page. This web page really is not MSNBC, it is really a web page that's controlled or compromised by the attacker, and all it does is it downloads a piece of malware to your PC. That's all it does.

Once that's done the attacker doesn't have to try to guess that you use Bank of America and just get you to type in your Bank of America password. Once your computer is infected with malware, regardless of which bank you go to or which other site you go to, your keystrokes will get logged and sent off to the attacker. So I believe malware is a very significant threat and the bad guys are getting very smart about how they, for instance, combine phishing together with social engineering in conducting such attacks.

Let me just tell you a little bit about some of the things you can do to protect yourself. I can of course go into more details off-line. But if you're using a home router you should change the default router password. You should also not use WEP, you should use a protocol called

WPA. You should use a personal firewall. You should always keep it on. You should not shut it off even for a couple of minutes because there's malicious machines that are continuously scanning the internet. You should use good anti-virus software. You can go to Pack.Google.com, for instance, and download anti-virus software for free. So you don't even have to pay anything, but use a good anti-virus package. If you get an alert that says hey, there's an update that needs to be done to your Windows software, click on it right away. You might be doing a lot of things, but click on it right away because what it means is that there's some vulnerable software on your computer and Microsoft is trying to help you patch that very quickly.

If your anti-virus package ever says that there is an update, you should definitely click on that right away because what's happened is there very well might be a vulnerability in your anti-virus package and the attackers can target that directly. So install those updates.

We can do a lot with security, but there might just be a case where your computer does get taken. Be sure to make backups or use a backup service.

What you can do also is you can use a browser with malware and phishing protection built in. For instance, Firefox 3 is a browser that is recommended to use. Also Google released a browser about a week or two ago called Chrome. Chrome also has malware and phishing protection. Basically that list of malicious web sites that may infect your computer with malware, Google has provided that list to popular browsers like Firefox 3 and Chrome so that you can basically be protected. Chrome is really cool. It has a lot of other advances. I'd recommend checking out Chrome if you haven't done so already.

Never install software that you don't trust, regardless of how cool that freeware show or package looks. Don't install it. The way operating systems work these days is when a program comes down to your PC it can pretty much do anything. When you go to financial sites, use your own bookmarks. Don't try to type the URL each time because you could mistype something and end up at a site that's really not your bank or your brokerage.

When you're connecting to banks and brokerages, use SSL, look for the little lock at the bottom of the browser. Don't ignore security warnings.

Use good passwords. A good password should not be your dog's name because your dog's name is probably in a

dictionary somewhere or probably on a list of common names and the attackers when they try cracking in, they don't try one passwords at a time. They have programs that try millions of passwords within just a couple of seconds. So they can go through long lists of passwords very quickly. So choose good passwords.

Choose also good reset questions. How many of you heard that Paris Hilton's phone got hacked back in 2004? The way that happened is her password reset question was what is your pet's name? So basically if I'm the attacker I enter P Hilton into T-Mobile.com or whatever. I say I forgot my password. The web site asks me for my dog's name. How do you get Paris Hilton's dog's name or pet's name? You do a search on Google or Yahoo and then basically you've got Paris Hilton's account.

If you use credit card numbers use one with a limit. There's also virtual one-time credit card numbers that some banks make available. Citibank makes something like this available.

In general, if you get an e-mail from a Nigerian prince with an offer that's too good to be true, it probably is.

My hope is that we can all work together as a community. The fact that I have to give a dozen rules to follow tells me that as a technologist we have a lot of work to do. We like to work together so that the situation does improve. My hope is that if we can all work together we can have the internet continue to grow into a platform that will allow us to communicate, collaborate, and conduct commerce in ways that as of yet we potentially haven't even imagined.

Thanks.

[Applause].

With that, I'd be happy to take any questions.

Question: I've got a quick question for you. How is Google and Yahoo and the other service providers, are they building a consortium to constantly, if you will, protect us?

Dr. Daswani: All these companies are working together. The way to look at this is that if users are getting attacked and there's bad things happening in the on-line commerce world, it's actually bad for everybody. We don't want people to have bad experiences, right?

For instance, in addition to us doing the malware blacklisting on search engine results, Yahoo has also announced that they're going to be doing it. In some cases there is data sharing in between such providers. My hope is that we can continue to do more and more of that.

Question: When I'm in Google and I'm in my account, and I'm e-mailing a person and I use a name, how come I get down the side a site suggested that has that name [inaudible]? And how does it happen so quickly and why is it happening? I don't think I see that on Yahoo.

Dr. Daswani: The question is a more general one about say using G-mail in particular. When you wrote e-mails, when you read e-mails, you see some ads to the right. Basically there's automated programs at Google that want to help you in various ways, so what they do is they, in an automated fashion, look up various words in the e-mails and try to find ads that are relevant to you and helping you solve problems that you have. So with regards to how does it happen so fast, in the same way that when you enter a search on Google you get back ads on the right side as well as search results very quickly, the same thing happens with mail.

So it's there to help identify advertisers that might have common interests with you.

Question: You talked about ways to defend. Are there any techniques you're aware of or you can share with us that allow us to step out of the fortress and go on the attack?

Dr. Daswani: I talked a little about defense, and the question is are there things we can do to attack? Absolutely.

For web sites, for instance, you shouldn't just write the code and then assume that it's passed your test and you're putting it out on line and it will work. You should always do what are called vulnerability assessments. So you should have say a third party company try to attack your site and/or use a service that does automated penetration testing of your site so that it can identify vulnerabilities. That's one thing that can be done. That's very useful.

At the same time, when you do a penetration test like that in an automated fashion, it's throwing test traffic, test attack traffic at your site. And that test attack

traffic may or may not be representative of the attacks that the bad guys are doing at this moment.

So it might be a good idea to for instance use tools like web application firewalls and/or other types of tools that don't just look at the windows and the doors on the side and try to identify vulnerabilities, but that also kind of serve as a watch guard that are looking at the real time traffic that's coming into the site 24x7.

Question: As a Google product security manager how are you lashed up with Google's legal department so as you're doing activities and the legal department is reacting and keeping Google out of trouble?

Dr. Daswani: The question is as a product security manager how do we work with legal. We work very closely with legal. We work in proactive ways with them to, for instance when we're developing a new product it's very important to, as they're developing the end user licensing agreement to make sure that we can support that and it makes sense from various angles, including security.

In other ways, when various companies get attacked, for instance, it's important for us to work with legal to figure out how should we go about trying to find the bad guys? How do we work with them? So we work with legal on many different fronts.

Question: How does legal go outside Google to work with federal officials or agencies to work on cyber crime?

Dr. Daswani: Actually it's not just legal that goes out to other agencies, but we have investigation teams where Google has hired a number of ex-secret service, ex-FBI types of folks that help us interface with those communities. So in the case of real attacks, we have all the appropriate connections to follow up and work with the appropriate authorities when that is necessary.

Question: As a company that [inaudible], how is Google being proactive to ensure that [inaudible]?

Dr. Daswani: The question is given that Google stores a lot of information, how is Google being proactive about protecting that user information?

There's a lot that Google is doing. For instance, with regard to even data about searches that come in, Google recently announced that Google will anonymize log data - initially we were anonymizing log data after 18 months and Google recently announced that we're going to

anonymize that log data after nine months, ahead of all our competitors. So we're leading and we're working to ensure user privacy.

Question: [Inaudible] ability within international law for you to have the ability to track down sources and perpetrators, malware and this sort of thing [inaudible] outside our borders? Where's the line? What can you do about it? What can't you do about it?

Dr. Daswani: The question is what can we do and what can't we do with regards to working with international authorities?

Depending upon the particular situation, it may make sense for us to work with the authorities here who then of course have appropriate connections internationally. At the same time, Google does have multiple offices around the world. In certain cases it might make sense for them to reach out to those authorities.

With regards to finding the bad guys in Russia and China, these things are challenging. There haven't been as many cases where the bad guys are found, and even if they're found and even if you can isolate who it is, the question is how do you actually prosecute them?

I think these are areas where Google will have to continue collaborating with our government, with the authorities, and with international authorities to make an impact on. I would love to see more cases where appropriate cyber criminals are brought to justice so that it can serve as a deterrent, as a message. But currently the cyber criminals are not scared, and we need to do a much better job on that front as a community. Google is happy to work with authorities to make that happen.

Question: I saw a lot of people taking notes. Is your presentation available to --

Dr. Daswani: I can make it available on-line. I'll make it available at NeilDaswani.com.

Question: Thank you very much for coming.

Dr. Daswani: Thank you for the invitation.

#