[IN]SECURE

OPEN. INFORMATIVE. TO THE POINT. Issue 33 - February 2012

INTERVIEW: FACEBOOK CSO

SECURING ANDROID: THINK OUTSIDE THE BOX

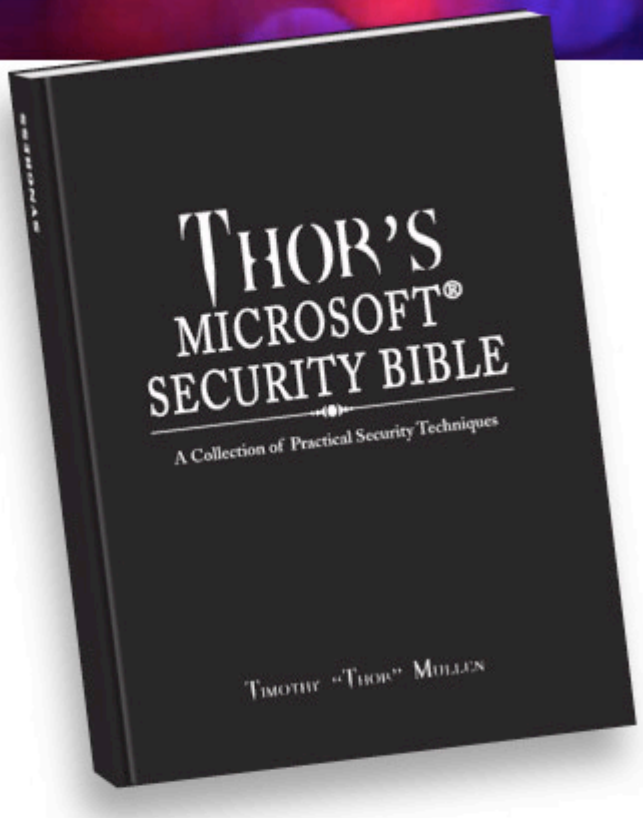
METASPLOIT: THE FUTURE OF PENETRATION TESTING WITH HD MOORE

USING AND EXTENDING THE VEGA OPEN SOURCE WEB SECURITY PLATFORM

SYNGRESS

Delve Deeper Into Security

with Syngress



Thor's Microsoft Security Bible

A Collection of Practical Security Techniques

By Timothy "Thor" Mullen

978-1-59749-572-1 | August 2011
Hardback | 322 pp.
$59.95 | €42.95 | £36.99 | $74.95

World-renowned security expert, Timothy "Thor" Mullen, presents a fascinating collection of practical and immediately implementable Microsoft security techniques, processes and methodologies uniquely illustrated through real-world process examples!

Securing the Cloud

Cloud Computer Security Techniques and Tactics

By Vic (J.R.) Winkler

978-1-59749-592-9
April 2011
Paperback | 290 pp.
$59.95 | €42.95
£36.99 | $74.95 AUD

The first book that helps you secure your information while taking part in the time and cost savings of cloud computing!

Security Risk Management

Building an Information Security Risk Management Program from the Ground Up

By Evan Wheeler

978-1-59749-615-5
May 2011
Paperback | 340 pp.
$49.95 | €35.95
£30.99 | $62.95 AUD

The definitive guide for building or running an information security risk management program.

syngress.com

TABLE OF CONTENTS

With this issue of (IN)SECURE Magazine, we enter our seventh year of publication. This time around we focus on Android security, we bring you the thoughts of the Facebook CSO and THE man behind Metasploit. To top if off, there are articles on web security, shellcode, mobile security, and more!

February is going to be a busy time for every information security company. The monumental RSA Conference is opening its doors later this month, and we'll be there to cover all the news and meet with companies and readers. I'm looking forward to the expo floor safari, there's always interesting technologies to discover. Look out for our camera, you might just be featured in an upcoming issue!

Mirko Zorz
Editor in Chief

Visit the magazine website at www.insecuremag.com

(IN)SECURE Magazine contacts

Feedback and contributions: Mirko Zorz, Editor in Chief - mzorz@net-security.org
News: Zeljka Zorz, Managing Editor - zzorz@net-security.org
Marketing: Berislav Kucan, Director of Operations - bkucan@net-security.org

Distribution

Online scam susceptibility of American consumers

PC Tools, in collaboration with the Ponemon Institute, announced the findings of its online scam susceptibility study of 1,858 American consumers.

The results of the survey show that close to half of US respondents think that they would be likely to provide personal or financial information online in each of the test scenarios presented:

Test Scenario	Rate %
An online prize	55%
Free antivirus software	53%
Get rich quick opportunity	53%
Free movie	48%
Online shopping registration	46%
Online donations	31%

The survey results also indicate that certain demographic groups are more susceptible than others. For example, respondents who indicated they are Independent supporters are the most susceptible to online scams, while supporters from the Green Party are the least.

Regionally, respondents who indicated they are from the Southwest are the most susceptible, while respondents from the Midwest and Pacific are the least.

The survey results also indicated that respondents from the following demographics are more susceptible to online scams:

- 18-25 year olds
- Females
- Less than a high school diploma
- Household income of $25,000 - $50,000
- Reside in the Southwest.

Unfortunately, many consumers don't realize that some online scams don't involve malware.

Traditional internet security is essential to maintain protection against viruses or malicious files and websites, but cybercriminals are changing their methods by tricking consumers into revealing their personal information, so this requires a very different protection approach.

Brazen Brazilian hackers opening cybercrime schools

Brazilian hackers are known for their preference for stealing and misusing phished banking credentials and credit card numbers, but also for their penchant to openly brag online about their illegal activities.

This relaxed attitude regarding the possibility of getting caught and tried for their illegal actions is due to the country's extremely inadequate anti-cybercrime laws, explained Kaspersky Lab's Fabio Assolini, who recently spotted another business venture initiated by the criminals.

"To help new 'entrepreneurs' or beginners interested in a life of cybercrime, some Brazilian bad guys started to offer paid courses," he revealed. "Others went even further, creating a Cybercrime school to sell the necessary skills to anyone who fancies a life of computer crime but lacks the technical know-how."

A number of different courses are offered, and while some seem like legitimate ones - how to become a designer, a Web designer, a hacker, a programmer - other not so much as they offer to teach how to become a "banker", a defacer or a spammer.

The courses can be bought online but - as unbelievable as it sounds - aspiring cybercriminals can also attend real-world classes at a location that is shared freely and, obviously, without any fear of law enforcement reactions.

Hackers steal $6.7 million in bank cyber heist

A perfectly planned and coordinated bank robbery was executed during the first three days of the new year in Johannesburg, and left the targeted South African Postbank - part of the nation's Post Office service - with a loss of some $6.7 million.

According to the Sunday Times, the cyber gang behind the heist was obviously very well informed about the post office's IT systems, and began preparing the ground for its execution a few months before by opening accounts in post offices across the country and compromising an employee computer in the Rustenburg Post Office.

Once the offices were closed for the New Year holidays, the gang put their plan in motion. They accessed the computer from a remote location and used it to break into Postbank's server system and transfer money from various accounts into the ones they opened.

Having also raised the withdrawal limits on those accounts, money mules had no problem withdrawing great amounts of money from ATMs in Gauteng, KwaZulu-Natal and the Free State during the next few days, stopping completely when the offices were opened again on January 3.

Unfortunately, the Postbank's fraud detection system hasn't performed as it should, and the crime was discovered only after everyone returned to work after the holiday break. Apparently, it should not come as a surprise - according to a banking security expert, "the Postbank network and security systems are shocking and in desperate need of an overhaul."

Mozilla offers alternative to OpenID

Mozilla has been working for a while now on a new browser-based system for identifying and authenticating users it calls BrowserID, but its only this January that all of its sites have finally been outfitted with the technology.

Mozilla aims for BrowserID to become a more secure alternative to OpenID, the decentralized authentication system offered to users of popular sites such as Google, Yahoo!, PayPal, MySpace and others.

"Many web sites store extensive user data and act on behalf of the user. While the browser may be fully under the user's control, many of the services that users enjoy are not. Sometimes, these web services handle data in ways that are of questionable value to the user, even detrimental," says Ben Adida, Mozilla's Tech Lead on Identity and User Data.

"It's clear that Mozilla needs to step up and provide, in addition to the Firefox browser, certain services to enhance users' control over their online experience and personal data."

Apart from BrowserID, Mozilla is also looking to launch Boot to Gecko (B2G), a standalone mobile web-based operating system, and an app store.

Stratfor hack exposes UK, US and NATO officials to danger, phishing

During the last days of 2011, Anonymous attacked Stratfor, a US-based research group that gathers intelligence and produces political, economic and military reports that help government organizations and major corporations asses risk.

Among the data they have managed to steal from its servers were names, home addresses, credit card details and passwords of Stratfor clients, 17,000 of which they have immediately shared with the public in order to prove the veracity of their claims.

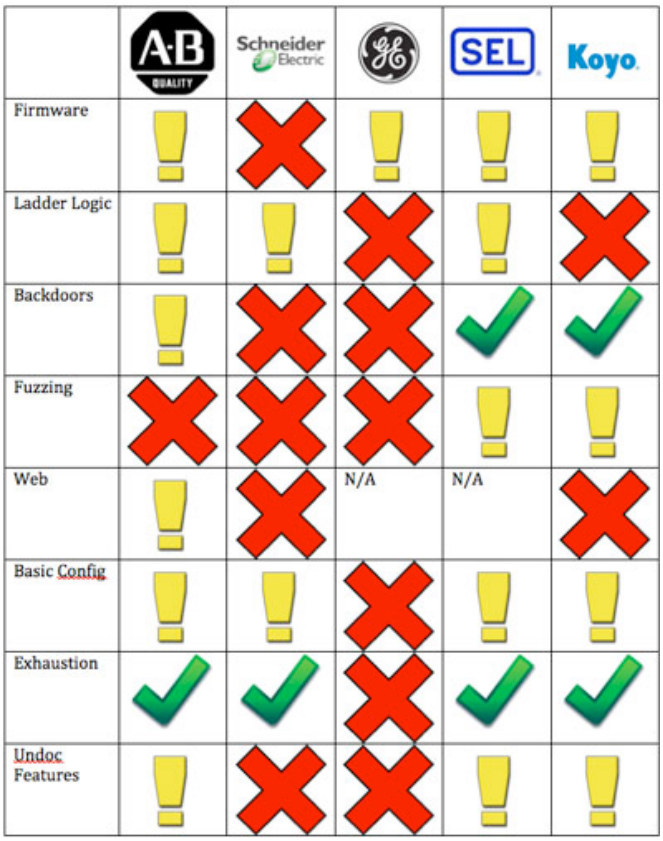
All in all, the hackers said that they have managed to put their hands on around 860,000 usernames, emails, and hashed passwords; internal emails and documents exchanged and worked on by the organization's employees; and around 75,000 credit card details complete with security codes required for no card present transactions.

The Guardian has hired cyber-security expert John Bumgarner to rifle through the information already leaked by the hacker group, and he has ascertained that thousand of emails and passwords belonging to UK, US and NATO officials were thusly made public.

19,000 email addresses and passwords and other personal data belonging to US military personnel were revealed, as well as those of seven officials of the UK's Cabinet Office, 45 of the Foreign Office, 14 of the Home Office, 67 police officers of the London Metropolitan Police and other officials, two employees with the royal household, 23 workers/members of the Houses of Parliament, and a number of intelligence officers. 242 Nato staffers have also had their emails revealed.

British officials and the government are still not worried about the revealed information posing any threat to national security. To be sure, the revealed (easily decryptable) passwords are those used by Stratford customers to access the content offered by the think-tank and not their email accounts.

Researchers demonstrate tragic state of SCADA security



At the SCADA Security Scientific Symposium held in Miami, visitors had the opportunity to hear a damning presentation held by researchers grouped around Project Basecamp which revealed that their testing of six widely used programmable logic controllers (PLCs) resulted in the discovery of alarming security bugs that are mostly design flaws and (even!) features, and of the fact that some of them can't even take a probing without crashing.

One of the devices, the Control Microsystems' SCADAPack, bricked early on into testing.

The remaining five (General Electric's D20ME, Koyo's Direct LOGIC H4-ES, Rockwell Automation's Allen-Bradley ControlLogix and Allen-Bradley MicroLogix, Schneider Electric's Modicon Quantum, and Schweitzer's SEL-2032) displayed a dazzling array of back door accounts, old hardware and firmware, lousy security controls, configuration files easily obtainable by attackers, buffer overflow and remotely exploitable vulnerabilities, unexpected crashes, weak password implementation and authentication protection, and inability to upload custom firmware.

Despite the reservations of some security experts that have questioned the researchers' action of making this information public before sharing it with the vendors, most industrial control security experts are satisfied that someone has finally pointed out these things they knew for years.

"A large percentage of these vulnerabilities the vendor already knows about and has chosen to live with, so this is not news to them," commented Dale Peterson, CEO of SCADA security firm Digital Bond, which organized the project, and said that the best way to avoid uncomfortable disclosures is to do a better job making secure products.

He expressed his belief that this presentation should be the moment when SCADA systems and PLC vendors finally realize that they have to take security more seriously. For their part, the researchers collaborated with Rapid 7 and Tenable in order to create test modules for the Metasploit Framework and the Nessus scanner for these vulnerabilities, in the hope that vendors will be pushed to make changes with security in mind.

Qualys expands its FreeScan service

Qualys announced its new and improved FreeScan service (freescan.qualys.com) to help SMBs audit and protect their web sites from security vulnerabilities and malware infections.

The new FreeScan service allows SMBs to scan their web sites for of malware, network and web application vulnerabilities, as well as SSL certificate validation, helping web site owners identify risk before hackers do in order to prevent data beaches and protect online visitors from infections.

Pwn2Own 2012: Changed rules, bigger prizes, no more mobile hacks

Pwn2Own, one of the most anticipated hacking contests that takes place each year at the CanSecWest conference in Vancouver, British Columbia, is set to unfold under dramatically different rules this year.

First and foremost, smartphone hacking is no longer on the table. This year edition will also reward the three most successful participants with cash prizes of $60,000, $30,000 and $15,000, respectively (plus the laptops they manage to compromise).

Also, a successfully compromised target will not be pulled from the competition as in previous years. All contestants can attack all targets during the whole three days of the contest, and the contest will be point-based.

"Any contestant who demonstrates a working 0day exploit against the latest version of the browser will be awarded 32 points," say the rules. "When the contest begins we will be announcing 2 vulnerabilities per target that were patched in recent years. The first contestant (or team) who is able to write an exploit for the announced vulnerabilities will be awarded 10, 9, or 8 points depending on the day the exploit is demonstrated."

For exploiting the already known vulnerabilities, contestants will only have to overcome DEP, and don't have to escape from a sandbox or protected mode. The browsers will be installed on Windows XP and Snow Leopard, and their versions will be made public at the beginning of the contest.

For the zero-days, hackers will be targeting browsers on fully patched Windows 7 and Mac OS X Lion machines. Also, one requirement that contestants must fulfill in order to win is to demonstrate at least one zero-day vulnerability on one of the targets.

As the in the previous year, Google is offering special prizes for Chrome "ownage": $20,000 for a set of bugs present only in Chrome that allow full unsandboxed code execution, and $10,000 for a compromise that used bugs both in Chrome and the OS for the same type of code execution.

Entrust Discovery now offers Microsoft CAPI query capabilities

Entrust expands its certificate discovery solution, Entrust Discovery, by broadening search capabilities for digital certificates residing within Microsoft's Cryptographic APIs (CAPI). And now with more than 25 basic or custom policy alert fields, Entrust Discovery offers stronger compliance tools.

"Understanding that today's organization often manages complex certificate environments, we provide more methods of discovering certificates and enhance the policy options once under management," said Entrust President and CEO Bill Conner.

Entrust Discovery assists organizations in gaining a complete perspective of deployed certificates. The solution finds, inventories and manages digital certificates across diverse systems to help prevent outages, data breach and non-compliance.

The solution now offers more policy alert fields, including issuer DN, expiry status, subject DN, key (e.g., RSA 2048), time valid, subject alt names (SAN) and certificate signature method.

Targeted attacks will change the economics of security

European Justice Commissioner, Viviane Reding, unveiled the new European Privacy Directive, designed to safeguard personal, identifiable information that is stored by private and public sector organizations.

All 27 European member states will be governed by the new rules, which could see companies being fined 2 per cent of global turnover if their customers' privacy is breached.

Under the new rules, all UK companies that suffer a security breach will have to inform the Information Commissioner within 24 hours of discovering a breach. Companies with more than 250 employees will have to appoint a privacy officer.

Corporations risk being fined up to 2 per cent of their global turnover for failure to adequately secure citizens' information. In addition, in a new "right to be forgotten" ruling, customers can request details of the information that companies hold about them and ask for it to be amended or removed.

Bruce Green, Chief Operating Officer at M86 Security, commented: "While we applaud the move to strengthen safeguards around individuals' private information, we recognize that this harmonization of data privacy rules across Europe will increase the data management overhead for companies of all sizes. The prospect of being fined two per cent of turnover will change the economics of security, because the cost of compliance compared to the financial risk of a breach will now fall firmly in favor of security for global enterprises. This will make information security a discussion for the boardroom, not just the domain of compliance specialists and privacy officers."

Symantec advises customers to stop using pcAnywhere

In a perhaps not wholly unexpected move, Symantec has advised the customers of its pcAnywhere remote control application to stop using it until patches for a slew of vulnerabilities are issued. According to a company white paper, the risks for the users are the following:

- Man-in-the-middle attacks (depending on the configuration and use of the product) because of vulnerable encoding and encryption elements within the software.
- If the attackers get their hands on the cryptographic key they can launch remote control sessions and, thus, access to systems and sensitive data. If the cryptographic key itself is using Active Directory credentials, they can also carry out other malicious activities on the network.
- If the attackers place a network sniffer on a customer's internal network and have access to the encryption details, the pcAnywhere traffic - including exchanged user login credentials - could be intercepted and decoded.

The white paper also contains security recommendations for minimizing the potential risk of using the software, since some customers cannot stop using it because its of critical importance to their business.

Martin McKeay, Security Evangelist at Akamai Technologies, pointed out that most remote desktop applications are directly exposed to the Internet because they are used by service providers for troubleshooting their clients' network equipment, and that that is unlikely to change in the near future.

MIS TRAINING INSTITUTE'S

INFOSEC WORLD

CONFERENCE & EXPO 2012

Over 70 Sessions to Help Solve Your Security Challenges:

- End-to-End Security for the Cloud Era
- Free Vulnerability Tools to Audit Security
- Mobile Banking: Securing the Next Financial Revolution
- Building a Web Application Security Assessment Program on a Budget
- Top 10 Windows Security Controls… and How to Correctly Collect Them
- Managing Sensitive Data in SharePoint
- Using Free Tools to Secure your Wi-Fi Network
- Pen Testing the Virtual Environment
- Using the Internet as an Investigative Tool
- iPhone and iPad Forensics
- Hacking and Defending MS SQL Server
- Privacy and Security Legal Update
- Identity Management For A New Era of Technologies
- MDMs Live! Helping IT Control Risky Androids and iPhones
- Protecting Against Malware on Mobile Platforms
- And much more…

Earn up to 54 CPEs!

April 2-4, 2012 ▪ Orlando, FL

Disney's Contemporary Resort

Optional Workshops:
March 31, April 1, 4, 5 & 6

CO-LOCATED SUMMITS:

CISO Executive Summit

Cloud Security Summit

IT Audit Management Summit

KEYNOTE SPEAKERS

Prof. Eugene H. Spafford, Ph.D.
Executive Director, CERIAS (Center for Education & Research in Information Assurance & Security), Purdue University

Nick Selby
Police Officer, DFW-Area; Co-Founder, Police-Led Intelligence

Mike McConnell
Executive Vice President, Booz Allen Hamilton; Former United States Director of National Intelligence; Vice Admiral, United States Navy, Ret; Former Director, National Security Agency

Dave Kennedy
CISO, Diebold Incorporated; Author of Metasploit: The Penetration Testers Guide and the Social-Engineer Toolkit

www.misti.com/infosecworld

Follow @InfoSec_World on Twitter

MIS TRAINING INSTITUTE — The International Leader in Audit & Information Security Training

PLATINUM SPONSOR: Entrust, ORACLE

GLOBAL EDUCATION SPONSOR: (ISC)²

ASSOCIATION SPONSORS: iapp, ISSA, INFRAGARD

The popularity of Android-based devices is driving their increased adoption in enterprise mobile applications, where security is a significant concern. In addition, designers of embedded systems are considering using Android for all forms of human-machine interfaces (HMI) in practically all major industries—automotive center stacks, medical device graphical interfaces, and home smart energy management panels, just to name a few.

Android brings to electronic products the power of open source Linux augmented with the graphical interfaces and app store infrastructure of one of the world's most popular mobile operating systems.

In addition, the rapidly emerging market for Android Mobile Device Management (MDM) solutions provides developers with the promise of a world-class remote device management infrastructure that can seamlessly tie into traditional back-end IT systems. MDM functions include remote monitoring and auditing, firmware updates, application configuration management and control, data-at-rest encryption, VPN services, remote wipe (e.g., when an embedded device is believed to be compromised), and more.

This article discusses the challenges and solutions for improving the security of Android-based devices in order to make them more suitable for enterprise, government, and other mission-critical environments.

Android security retrospective

As part of Android's original introduction in 2008, Google touted improved security in its smartphones. Google's website (code.google.com/android) lauded the platform's security: "A central design point of the Android security architecture is that no application, by default, has permission to perform any operations that would adversely impact other applications, the operating system, or the user." Days after the release of the first Android phone, the G1, a well-publicized, severe vulnerability was found in the phone's Web browser. But the G1's security woes didn't end there.

In November, hackers discovered a way to install arbitrary programs on the phone, prompting this lament from Google: "We tried really hard to secure Android. This is definitely a big bug. The reason why we consider it a large security issue is because root access on the device breaks our application sandbox."

In fact, the Android bug would silently and invisibly interpret every word typed as a command, and then execute it with superuser privileges.

In late 2010, security researchers uploaded to the Android market a spoofed Angry Birds game application that surreptitiously downloaded other apps without the user's approval or knowledge.

The extra downloads were malicious, stealing the phone's location information and contacts, and sending illicit text messages. As part of their work, the researchers reported numerous weaknesses in Android, including a faulty use of SSL, a lack of application authentication, an easy method of breaking out of the Android Dalvik virtual machine sandbox via native code, and the focus of the attack—a weak permissions architecture.

Next, we visit our favorite website, the U.S. CERT National Vulnerability Database. A search on Android turns up numerous vulnerabilities of varying severity. Here is a sampling of the worst offenders:

• CVE-2011-0680: Allows remote attackers to read SMS messages intended for other recipients.
• CVE-2010-1807: Allows remote attackers to execute arbitrary code.
• CVE-2009-2999, -2656: Allows remote attackers to cause a denial of service (application restart and network disconnection).
• CVE-2009-1754: Allows remote attackers to access application data.
• CVE-2009-0985, -0986: Buffer overflows allow remote attackers to execute arbitrary code.

We point out these particular vulnerabilities because they fall into the most serious severity category of remote exploitability.

These vulnerabilities are specific to the Android stack that runs on top of Linux. Android is, of course, susceptible to Linux kernel vulnerabilities as well. The rapid development and monolithic architecture of Linux has been well publicized. Lead Linux kernel authors have published multiple installments of a Linux kernel development statistical overview, and the numbers are staggering.

With 20,000 lines of code modified per day, 6,000 unique authors, and rapid growth in its overall code base, it should come as no surprise that dozens of Linux kernel vulnerabilities are reported each year, and that a steady stream of undiscovered vulnerabilities are latent in every Linux distribution deployed to the field.

While a significant portion of the growth and churn in the Linux kernel code base is due to the continual adding of support for new microprocessors and peripherals, the core kernel itself, including networking and file system support, also undergoes rapid change.

CVE-2009-1185 documents a flaw in the Linux netlink socket implementation, and is but one example of a Linux vulnerability that has allegedly been used to compromise Android devices. CVE-2009-2692, informally known as the proto-ops flaw, is a set of bugs in the Linux kernel's management of file and network access objects.

A trivial user mode program can be used to subvert an Android system using this vulnerability. The proto-ops flaw was latent in the Linux kernel for eight years before researchers discovered it.

Because its architecture for kernel object management is so entrenched, Linux remains susceptible to the vulnerability as new device drivers and communication mechanisms are added to the code base.

Android device rooting

Android rooting (also known as jailbreaking) is the process of replacing the manufacturer-installed kernel (Linux) and/or its critical file system partitions. Once a device is rooted, the hacker can change Android's behavior to suit his or hers particular desires.

The term rooting originates from the UNIX concept of root privilege, which is needed to modify protected functions. The goals of Android hackers range from the hobbyist's desire to overclock a CPU for better performance (at the expense of battery life) and install custom applications, to more malicious pursuits, such as illegally obtaining carrier network services, and installing key loggers and SMS snoopers.

The collection of new and replaced files installed by the hacker is referred to as a custom ROM, another imperfect reference to the concept of firmware that is often deployed in read-only memory.

Android vulnerabilities are often used by hackers to root Android phones. The rate of vulnerability discovery is such that practically every Android consumer device has been rooted within a short period of time, sometimes within a day or two of release.

In addition to software vulnerabilities, secure boot problems are another major source of Android rooting attacks. Some Android device makers, such as Barnes and Noble with its Nook Color, have permitted (if not encouraged) rooting in order to facilitate a wider developer community and device sales.

In this case, rooting is usually accomplished with a form of side-loading/booting using an SD card or USB to host or install the custom ROM. The manufacturer-installed boot loader does not cryptographically authenticate the Android firmware, paving the way for ROM execution.

Some device makers have gone to great lengths to prevent rooting for various reasons. Obviously, many developers using Android will want to lock down the Android OS completely to prevent unauthorized modification and malicious tampering.

One of the most high-profile secure boot failures in this realm is the Amazon Kindle. The presumed aim of locking down the Kindle is to force users to access Amazon content and require use of the Kindle e-reader software. The Amazon secure boot approach attempted to authenticate critical system files at startup using digital signature checks. Hackers used vulnerabilities in Linux to circumvent these checks and run malicious boot code, rooting the device.

Yes, we paint a grim picture of Android security. However, the picture is based on a simple fact that shouldn't be surprising—Android was never designed to provide a high assurance of security.

ANDROID VULNERABILITIES ARE OFTEN USED BY HACKERS TO ROOT ANDROID PHONES

Mobile phone data protection: A case study of defense-in-depth

Android's tremendous popularity, juxtaposed with its lack of strong security, has sparked a rigorous scramble by software vendors, device OEMs, systems integrators, and government security evaluators to find ways to retrofit Android-based devices with improved system security.

One approach to raising the level of assurance in data protection within an Android-based device is to employ multiple encryption layers. For example, an Android smartphone can use a layer four (OSI model) SSL VPN client to establish a protected data communication session. An IPsec VPN application, running at layer three, can be used to create a second, independent connection between the smartphone and the remote endpoint (Figure 1).

This secondary connection uses independent public keys to represent the static identities of the endpoints. The data in transit is doubly encrypted within these two concurrent connections. This layered security approach is an example of defense-in-depth.

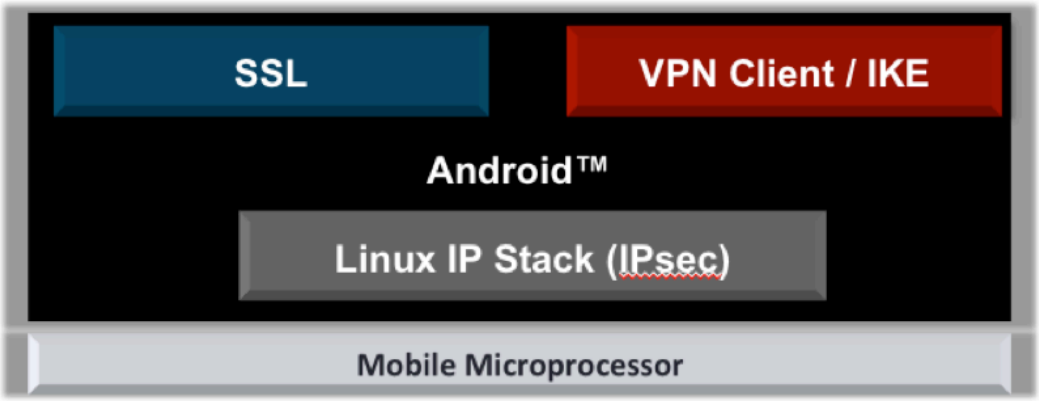


Figure 1 - Multiple layers of encryption within Android.

The concept of defense-in-depth originated in the military—multiple layers of defense, such as a combination of mines and barbed wire, rather than just mines or barbed wire alone, to increase the probability of a successful defense, as well as potentially to slow the progress of an attacker.

Defense-in-depth has been successfully applied in war since ancient times, and the concept is alive and well in the information security age.

Let's consider a few of the threats against an SSL data protection application. An attacker can attack the application directly, perhaps exploiting a flaw in the SSL software stack, to disable encryption entirely or steal the encryption keys residing in RAM during operation. An attacker can try to steal the static public SSL keys stored on disk. If these keys are compromised, the attacker can impersonate the associated identity to gain access to the remote client over a malicious SSL session.

Malware elsewhere in the Android system can use side channel attacks to break the SSL encryption and recover its keys.

Layered SSL/IPsec data protection is a sensible application of defense-in-depth to counter these threats. If an attacker is able to break the SSL encryption, the IPsec layer will continue to protect the data. An attacker may be able to steal the SSL keys but not the IPsec keys. The attacker may be able to install malware into the SSL application but not the IPsec application. The SSL application may exhibit side channel weaknesses to which the IPsec application is immune. To succeed, the attacker must break both the SSL and IPsec encryption layers.

Clearly, this layered approach depends on the independence of the layers. Most importantly, the SSL and IPsec private keys must be independently stored and immune to a single point-of-failure compromise. However, in a typical Android environment, both the SSL and IPsec long-term private keys are stored within the same flash device and file system. Furthermore, the key stores are not protected against physical attacks.

This environment provides numerous single points of compromise that do not require sophisticated attacks. A single Android root vulnerability or physical attack on the storage device can compromise both sets of keys and encryption layers.

The run-time environment must provide strong isolation of the SSL and IPsec application layers, and the run-time environment itself must not provide an attack surface through which to break that isolation. Much of the research and product development aimed at Android security has focused, in one form or another, on providing sandboxes for data isolation and the protected execution of critical functions. Those sandboxes are used to realize the layered encryption approach.

Let's now compare and contrast the various approaches for Android sandboxing. Developers considering the adoption of Android in their next-generation designs can use this comparison to make sensible security choices.

Android sandboxing approaches

Separate hardware

One sandboxing approach is to have multiple microprocessors dedicated to the differing tasks. While Android smartphone OEMs are unlikely to add additional hardware cost to their designs, custom electronic product developers may have more options depending on many factors, including form-factor flexibility.

For example, a PCI-capable design may be able to host an IPsec VPN card that wraps the second layer encryption around the main processor's Android SSL. In some cases, however, the additional hardware size, weight, power, and cost will be prohibitive for this approach.

Multi-boot

The multi-boot concept has been attempted on a handful of laptops and netbooks over the years. In a dual boot laptop scenario, a secondary operating system, typically a scaled-down Linux, can be launched in lieu of the main platform operating system. The scaled-down system is typically only used for Web browsing, and the primary goal is to enable the user to browse within a handful of seconds from cold boot. The secondary operating system resides in separate storage and never runs at the same time as the primary platform operating system. In some cases, the light-weight environment executes on a secondary microprocessor (e.g., an ARM SoC independent of the netbook's main Intel processor). On an Android mobile device, the primary Android can be hosted on internal NAND flash, and a secondary Android can be hosted on an inserted microSD card (Figure 2).

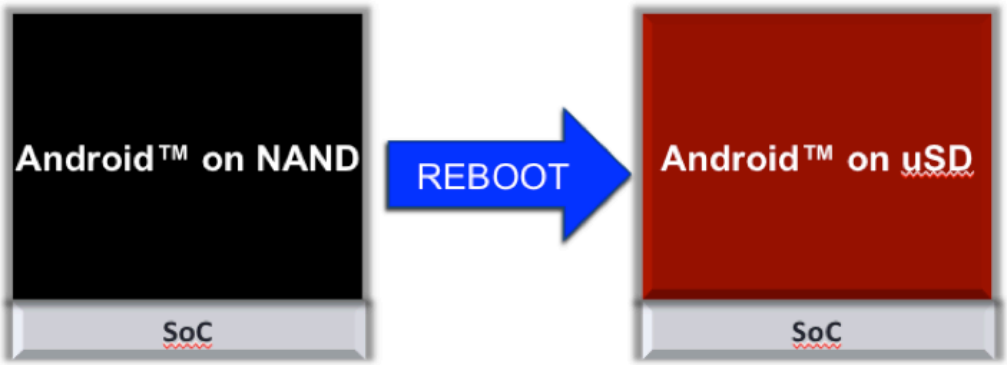


Figure 2 – Dual-boot Android.

The secondary operating system provides good isolation from a security perspective.

However, the inconvenience of rebooting and the inability to seamlessly switch between environments has severely limited adoption. The multi-boot option is also impractical for the layered encryption use case that requires concurrent execution of the sandboxes.

Webtop

The webtop concept provides a limited browsing environment (the webtop), independent from the primary operating system environment. However, instead of a dual boot, the webtop runs as a set of applications on top of the primary operating system.

In the case of the Motorola Atrix Android smartphone released in 2011, the webtop sandbox is an independent file system partition that contains a limited Ubuntu Linux-based personality (Figure 3).

The primary Android partition is located on the same internal NAND flash device within the phone. The Atrix webtop is intended to provide a desktop-like environment for users that dock the phone on a separately purchased KVM (keyboard/video/mouse) apparatus.

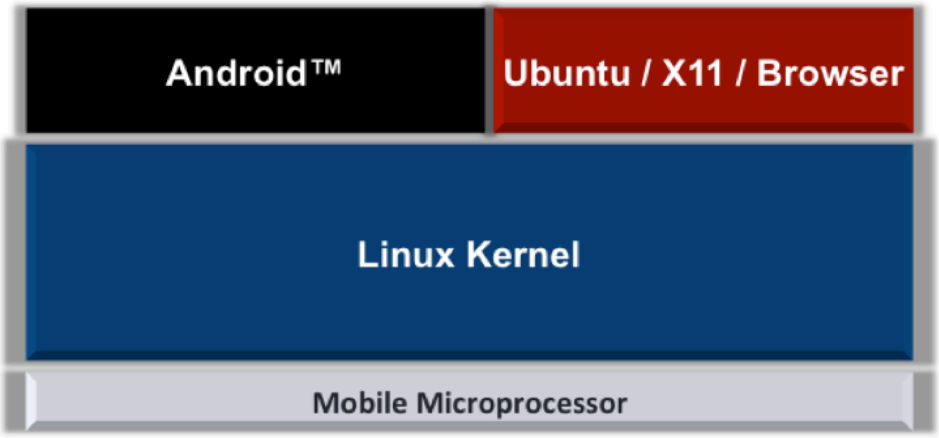


Figure 3 - Android webtop environment.

While webtop was most likely not intended as a security capability, one mapping of this approach to the layered encryption use case is to execute IPsec from the primary Android environment and an SSL-based Web session from the webtop sandbox.

The problem with this approach is that the entire Linux kernel, including its TCP/IP stack, is depended upon for the isolation of the webtop's SSL from the Android IPsec.

Mobile Device Management (MDM) encrypted containers

The growing popularity of Android mobile devices and the desire to use them in the workplace has spawned dozens of MDM products and companies. The two main purposes of MDM are to provide mobile data protection and IT management services.

Manageability includes application configuration (ensuring that all employees have an approved set of preloaded software), auditing, document management, and remote wipe (disabling the handset when an employee leaves the company).

Data protection covers both data at rest and data in transit (e.g.VPN to the corporate network).

Android MDM solutions often use application-level encryption. For example, an enterprise e-mail client may implement its own encryption protocol for the connection between a mobile device and an enterprise e-mail server, and its own encryption of the e-mail folders resident on the phone.

Some MDM solutions use Android profiles to divide the Android system into two sets of applications—one for the user's personal environment and one for the enterprise-managed environment (Figure 4).

When the enterprise profile is invoked, the MDM product may automatically turn on encryption for data associated with that profile. Numerous other Linux controls can be used to improve the isolation of profiles, including chroot jails and operating system-level resource grouping techniques like OpenVZ.

Clearly, this approach can be used to implement the layered encryption use case—the MDM application can create an SSL connection on top of the underlying Android's IPsec connection.

However, once again, the underlying Android operating system is relied upon for the security of both layers.

Remoting

One approach to enterprise data protection in Android is to not allow any of the enterprise data on the mobile device itself. Rather, the only way to access enterprise information is using a remote desktop and/or application virtualization. When the device is not connected to the enterprise (e.g. offline operation), enterprise applications and services are unavailable.

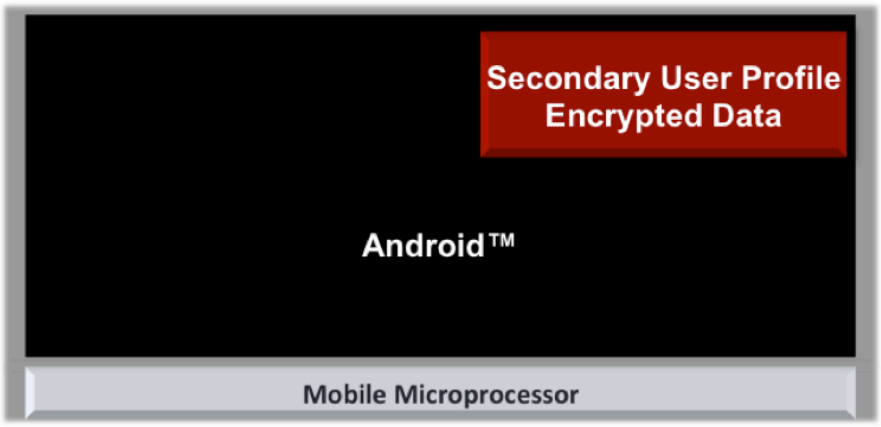


Figure 4 - MDM containers.

While the result is a neutered device that defeats the purpose of having such a powerful hardware platform with multiple cores and multimedia accelerators, there are certainly use cases that can take advantage of remoting.

Remoting precludes the requirement for local data protection; however, our use case for layered data-in-motion protection remains. The remoting application (Figure 5) provides SSL encryption while the underlying Android runs IPsec. Once again, the underlying Android operating system is relied upon for the security of both layers.

Type-2 hypervisor

Type-2 hypervisors are similar to webtops and MDM containers in that the secondary environment runs as an application on top of the primary operating system. However, instead of hosting only a browser, the secondary persona is a full-fledged guest operating system running within a virtual machine created by the hypervisor application (Figure 6).

The hypervisor uses the primary operating system to handle I/O and other resource management functions.

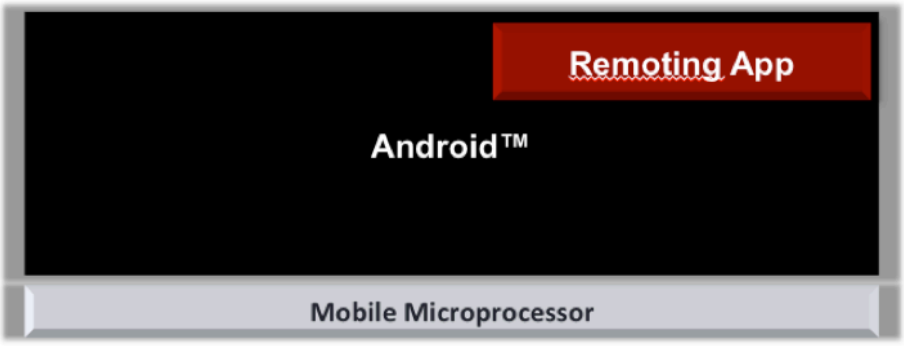


Figure 5 - Remoting.

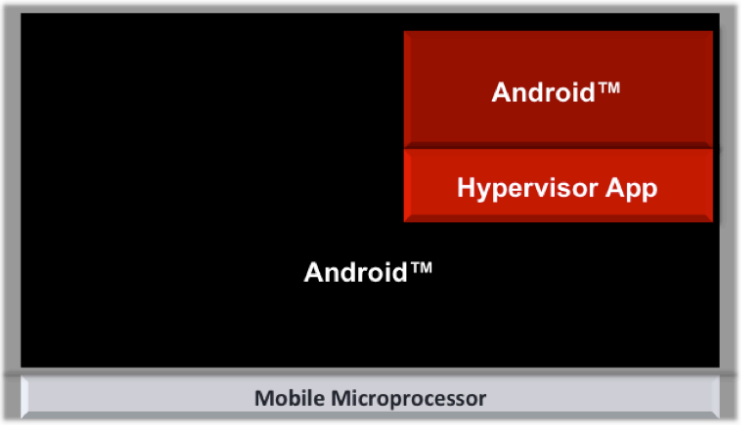


Figure 6 - Type-2 hypervisor.

Type-2 mobile hypervisor products, such as VMware MVP, are used to provide an enterprise management persona on top of the primary Android environment. The virtualized Android can use an SSL connection to the enterprise while the underlying Android's IPsec is also used to wrap the communication between endpoints.

However, once again, the Type-2 model fails to provide strong isolation. Faults or security vulnerabilities in the primary general-purpose operating system will impact the critical functions running in the virtual machine. Furthermore, Type-2 hypervisor applications deployed in the enterprise space have been found to contain vulnerabilities that break the sandbox.

Sandboxes built on sand

Constant reader, hopefully you observe as obvious the common weakness among all of the sandboxing approaches previously described. Multiple Android applications, MDM containers, remoting applications, webtops, and Type-2 hypervisors all attempt to retrofit security to the Android kernel itself.

The Android/Linux system, while providing rich multimedia functionality of which mobile and embedded designs can take good advantage, is riddled with security vulnerabilities that simply cannot be avoided. High-assurance security must be designed from the beginning.

But while high assurance cannot be retrofitted to Android itself, it can be retrofitted at a system level. Let's take a look at how.

Type-1 hypervisor

Type-1 hypervisors also provide functional completeness and concurrent execution of a secondary enterprise persona. However, because the hypervisor runs on the bare metal, persona isolation cannot be violated by weaknesses in the persona operating system. Thus, a Type-1 hypervisor represents a promising approach from both a functionality and security perspective. But the hypervisor vulnerability threat still exists, and not all Type-1 hypervisors are designed to meet high levels of safety and security.

One particular variant, the microkernel-based Type-1 hypervisor, is specifically designed to meet high-assurance, security-critical requirements. Microkernels are well known to provide a superior architecture for safety and security relative to large, general-purpose operating systems such as Linux and Android.

In a microkernel Type-1 hypervisor, system virtualization is built as a service on the microkernel. Thus, in addition to isolated virtual machines, the microkernel provides an open standard interface for lightweight critical applications, which cannot be entrusted to a general-purpose guest. For example, SSL can be hosted as a microkernel application, providing the highest possible level of assurance for this encryption layer. IPsec packets originating from Android are doubly encrypted with the high-assurance SSL layer service before transmission over the wireless interface (Figure 7).

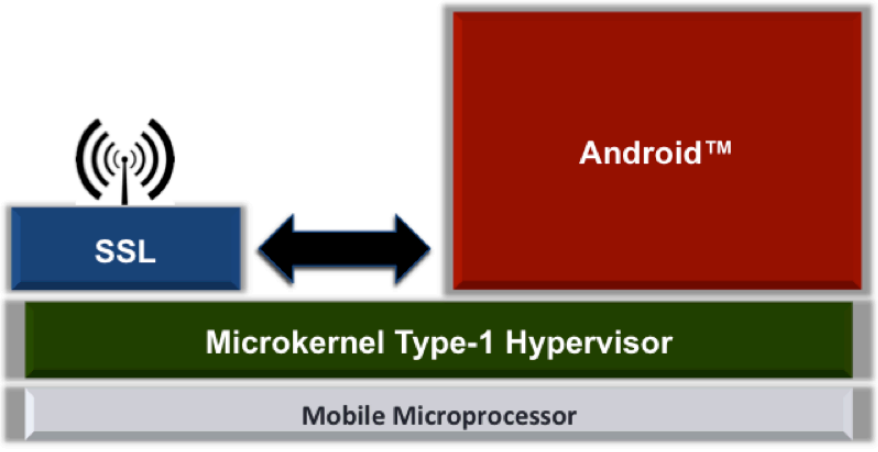


Figure 7 - Microkernel Type-1 hypervisor approach to layered data-in-motion encryption.

The real-time microkernel is an excellent choice for practically any mobile and embedded system since the microkernel can host any real-time application not appropriate for the Android/Linux environment.

The microkernel Type-1 hypervisor typically uses the microprocessor MMU to isolate the memory spaces of the primary Android environment and the native SSL encryption application. However, device drivers in Android may use DMA that can violate the memory partitioning by bypassing the MMU entirely.

Running the hypervisor in TrustZone on an applicable ARM-based microprocessor, using an IOMMU, or using the hypervisor itself to mediate all DMA bus masters are all potential approaches to guarding this attack vector.

The isolation properties of some secure microkernels can even protect against sophisticated covert and side-channel, software-borne attacks.

Physical security

Now that we have an approach that prevents software attacks from breaking the sandbox between protection layers, let's take defense-in-depth a step further and consider how the layered encryption system can be protected from physical attacks. For example, a lost or stolen mobile device in the hands of a sophisticated attacker is susceptible to memory snooping, power analysis, and other invasive and non-invasive physical attacks.

While physical protection of the entire device may not be practical, targeted physical protections can make a huge difference in overall system security. A secure element can be used to provide physical protection of critical parameters, including private keys. Several industry standards bodies are examining this requirement and offering solutions.

For example, GlobalPlatform (www.globalplatform.org) recommends the use of TrustZone, coupled with some form of secure element, to protect critical parameters and cryptographic functions used for mobile payments. The Trusted Computing Group (www.tcg.org) is working on the specification for a Mobile Trusted Module (MTM) that is comparable to today's Trusted Platform Modules (TPMs) found in laptops and PCs.

Most of the work being done in this area is in its infancy; full specifications are not complete, and commercial products that incorporate these standards are not yet on the market.

However, the concept of the MTM can be combined with the functionality of a smartcard to provide a mobile hardware root of trust with secure key store capability.

This approach offers a single element that can provide a secure trust anchor for secure boot and remote attestation, as well as a secure key store for device, user, and application keys and certificates.

For example, a smartcard chip can be incorporated into a microSD device and attached to a smartphone (Figure 8). This approach provides the physical security benefits of a secure element while allowing credentials to move with the user by removing and then inserting the microSD into another device.

Of course, implementations will vary depending on the types and sophistication of physical protections available. But a hardware-based root of trust enables a higher-level FIPS-140 certification and provides an important additional layer of security independent of the microkernel-based runtime environment isolation.

Summary

Layered encryption as a defense-in-depth strategy is a sensible approach to increasing the assurance of Android-based data protection services. However, it is not sensible to run both layers within the Android environment itself. There is simply too much vulnerability to prevent both layers from being simultaneously subverted. Designers considering Android must also carefully sandbox critical security functions outside of the Android system. Modern microprocessors and system software solutions provide the requisite features to get the best of both worlds—the power of Android's multimedia and applications deployment infrastructure alongside, but securely separated from critical system security functions.

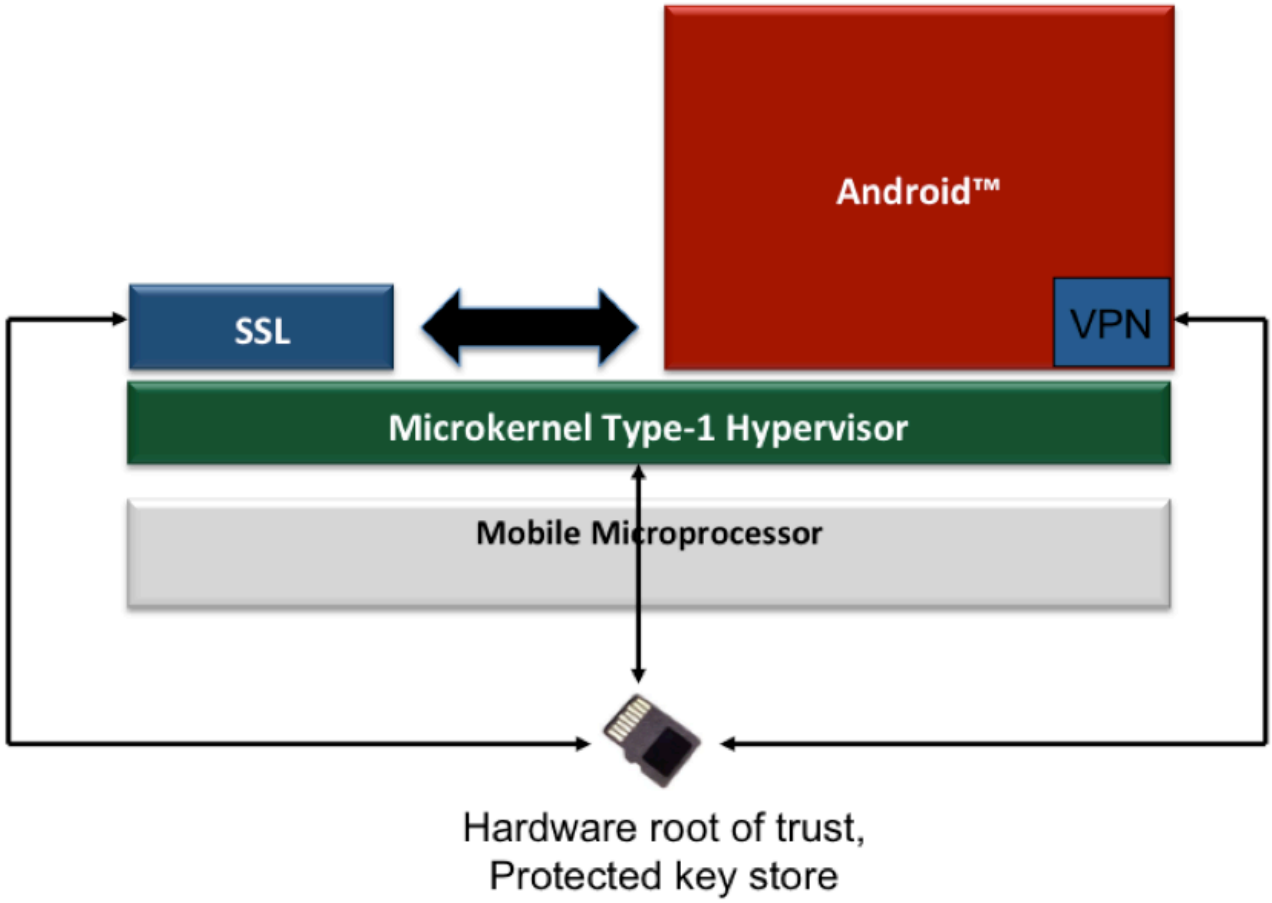


Figure 8 - Adding physical security protection via attached smartcard to the microkernel Type-1 hypervisor.

Kirk Spring is the VP of Technology for SafeNet. Currently he oversees SafeNet's strategic development of security solutions that includes technology sharing of both its commercial and government products. Mr. Spring earned his bachelor of science in Computer Engineering from Oakland University and has been with SafeNet since 2001. Prior to SafeNet, Mr. Spring was at Harris Corporation, Allied Signal Corporation, and Hughes Ground Systems.

David Kleidermacher is CTO at Green Hills Software where he is responsible for technology strategy, platform planning, and solutions design. Kleidermacher is a leading authority in systems software and security, including secure operating systems, virtualization technology, and the application of high robustness security engineering principles to solve computing infrastructure problems. Kleidermacher earned his bachelor of science in computer science from Cornell University and has been with Green Hills Software since 1991.

Joe Sullivan is the Chief Security Officer at Facebook, where he manages a small part of a company-wide effort to ensure a safe internet experience for Facebook users. He and the Facebook Security Team work internally to develop and promote high product security standards, partner externally to promote safe internet practices, and coordinate internal investigations with outside law enforcement agencies.

Being the CSO of Facebook certainly puts you into the spotlight. How have your prior positions prepared you for your work at Facebook?

I can think of two important ways my prior positions have helped prepare me for my current responsibilities. Before Facebook I worked as a federal prosecutor working on cybercrime cases that were in the media every day and then worked at eBay during the early part of the 2000s when that company was celebrated and scrutinized.

In both of those places I was challenged to develop creative solutions - because we were breaking new ground in areas where there was not much precedent. Likewise, in both I learned how to stay effective and focused even when under a serious microscope. Both skills, the ability to develop creative solutions to new and unique problems, and the ability to stay focused on addressing real risks and threats while under great scrutiny, are critically important for succeeding in my role at Facebook.

Facebook has partnered with the National Cyber Security Alliance on the STOP. THINK. CONNECT. campaign over two years ago. What are your thoughts on how public-awareness-raising campaigns can be improved in the future?

If you look at internet education safety campaigns before this effort by NCSA, you see a bunch of different parallel efforts focused on the same problems but using different tactics and terminology. This initiative is important because it brings together an incredibly wide

spectrum of technology, communication and other companies to work with government on developing unified messaging.

Having consistent terminology is critical to education in a complex area and with this effort the sum of our individual efforts working together is much greater than it would be if we invested the same in education but without this degree of coordination.

Facebook launched its bug bounty program in August last year and has already doled out quite a sum to outside security experts. Have there been any great surprises? Has the program influenced the way that the security team approaches code reviewing? Did you offer employment to a particularly successful bug hunter/are you thinking about doing it?

The program has been successful beyond our expectations. First, it really blew up the assumption that there are only a small number of quality researchers able and willing to report meaningful bugs. On the contrary, we have found that there is an incredibly vibrant entrepreneurial security community around the world that is passionate about engaging on web application security.

We have had submissions from over 16 countries and have already payed out over $150,000 in bounties. In the process we have built great relationships with some amazing researchers from every corner of the globe. And yes, we do have a summer intern coming who we met through the program.

I don't think it has influenced the way we review code, but it does make us feel even better about the overall review process we have in place being as complete as possible. We intend to keep investing in this program and are always looking for feedback on how to make it better.

Our latest iteration was to add a debit card as a payment option so that we can reload easily for people who submit bugs regularly.

We know that we will always be out-numbered by the bad guys, but we can overcome that by making sure that our systems are up to the challenge.

As the number of Facebook users grows seemingly exponentially, does your security team as well? What security-related problems currently give you the biggest headaches?

We do continue to grow in size, but we are also constantly challenging ourselves to develop in such a way that every employee focused on security has a greater individual impact tomorrow than that person did today. We can do that both by continuing to innovate on our approaches to security and investing in system and infrastructure.

We know that we will always be out-numbered by the bad guys, but we can overcome that by making sure that our systems are up to the challenge. An example of how things change and new headaches arise the sudden increase in what we call self-XSS during last year. Self-XSS attacks used social engineering to trick users into copying-and-pasting malicious javascript into their browser, thereby self-propagating the spam and evading our detection systems.

Before the attacks increased dramatically most experts would have doubted that a social engineering scheme could work at such scale.

Fortunately, we reacted quickly and have had success beating it back. In addition to improving internal detection mechanisms, we have worked with browser vendors to make it harder for spammers to take advantage of this vulnerability in the browser, and we have partnered with external companies to make our malicious link detection system more robust.

We are still battling this but thankfully it is much less of a headache than it use to be.

I can't remember the last time I saw a bogus or information-collecting app being pushed onto users by third party developers, and I recall them being plentiful at one point in time. How did you solve that particular problem?

We have several different teams that work closely together to ensure people have a great experience when connecting with applications that leverage our platform.

Major props go to the platform integrity engineers who have been constantly iterating on the automated systems that we put in place to secure our Platform. Of particular note were the changes we made last July which made significant improvement to the enforcement systems so we can identify and disable apps that violate our policies as quickly as possible.

These changes instituted granular enforcement which selectively disables an app's ability to propagate through Facebook based on the amount of negative user feedback - so that an app that has been reported for abusing chat will have this feature disabled until the developers have made substantial changes.

In the future, we are moving to more sophisticated ranking models where the amount of distribution will be a function to the app's quality. Good content will be seen by more people, while lower quality or spammy apps will be seen by fewer people or no one.

We believe this will reward apps that provide great experiences while minimizing the negative impact of poor quality apps.

Zeljka Zorz is the Managing Editor of (IN)SECURE Magazine and Help Net Security.

Are Hackers Finding a Way Into Your Network?

GFI LANguard

Award-winning vulnerability management software

To lower the security risk you need GFI LANguard, a solution that provides network vulnerability scanning, patch management and auditing in one integrated package. This award-winning solution allows you to scan, detect, assess and rectify vulnerabilities on your network faster and more effectively.

GFI

WEB & MAIL SECURITY
ARCHIVING & FAX
NETWORKING & SECURITY

Download your FREE trial version from **www.gfi.com/lannetscan/**

tel: +1 (888) 243-4329 | fax: +1 (919) 379-3402 | email: ussales@gfi.com | url: www.gfi.com/lannetscan/

White hat shellcode: Not for exploits

by Didier Stevens

The goal of this article is to plant a seed of the idea that shellcode has a place in your defense toolbox. I do not want to teach you how to write shellcode, neither do I want to present a complete anthology of white hat shellcode. What I want is to show a few examples in order to help you be more creative, so that when you are facing a problem in your IT security job, you will also consider shellcode as a potential solution.

When a system is attacked, be it by malware or by a human, shellcode is often involved. Shellcode is executed on the system to change its behavior, so that the system opens up to the attacker. But why couldn't you use shellcode to change the behavior of your system, too, so that it defends itself against an attacker? There is no reason why you couldn't do this.

As the administrator of the system, you have an advantage over the attacker. While the attacker has to rely on exploits that often offer no guarantee that the shellcode will execute, you, on the other hand, can use reliable methods to inject and execute shellcode. Shellcode is almost always used in attack scenarios, but it can also be used for defense.

Shellcode is a tool, and it can be a solution to your problem.

What is shellcode? Shellcode is a program, but it has some characteristics that differentiate it from applications like .exe files. Shellcode is a program that is location-independent and comes as a binary file without any metadata.

Example 1: Testing a security setup

In the first example, we will test our security setup with shellcode. People regularly ask me for malware so they can test their security setup. First, that is a bad idea, and second, you can do without. Why is using malware a bad idea? It is dangerous and not reliable.

Say you use a trojan to test your sandbox. You notice that your machine is not compromised. But is it because your sandbox contained the trojan, or because the trojan failed to execute properly? It might surprise you, but there is a lot of unreliable malware out in the wild - malware that will crash more often than not, malware that will flat-out refuse to run in certain environments, like virtual machines.

So how can you reliably test your sandbox without risking infection, or even worse, have malware escape into your corporate network?

You can do this with shellcode. Here is an example of simple shellcode that will create a file in the directory of your choice (This shellcode includes a library that is not discussed in this article):

```
segment .text
      call geteip
      geteip:
      pop ebx

      ; Setup environment
      lea esi, [KERNEL32_FUNCTIONS_TABLE-geteip+ebx]
      push esi
      lea esi, [KERNEL32_HASHES_TABLE-geteip+ebx]
      push esi
      push KERNEL32_NUMBER_OF_FUNCTIONS
      push KERNEL32_HASH
      call LookupFunctions

      ; CREATEFILEA and CLOSEHANDLE
      push 0x0
      push 0x80
      push 0x2
      push 0x0
      push 0x0
      push 0x0
      lea eax, [FILENAME-geteip+ebx]
      push eax
      call [KERNEL32_CREATEFILEA-geteip+ebx]
      push eax
      call [KERNEL32_CLOSEHANDLE-geteip+ebx]

      ret
```

Let us assume you sandboxed your preferred browser, Firefox, and now you want to test if Firefox is restricted from writing to the system32 directory.

For this, we use shellcode that creates file c:\windows\system32\testfile.txt and inject this shellcode in process firefox.exe.

If the test file was not created in the system32, you have successfully verified that your sandbox prevents Firefox from writing to the system32 directory. You can also start Sysinternal's procmon and look for “access denied” messages from Firefox. This is further proof that the shellcode tried to write to system32 but was denied.

This method is very reliable, especially compared with the use of real (unreliable) malware. If you need to test access to other resources, like the registry, you just need to use shellcode that writes to a particular key in the registry.

Example 2: Enforcing Permanent DEP

DEP is an important security feature introduced with Windows XP SP3. But not all applications use DEP, so here is how you can enforce it. DEP can be enabled by setting a flag in the executable file (the NO_EXECUTE flag) or by calling WIN32 API function SetProcessDEPPolicy.

SetProcessDEPPolicy has one advantage over the NO_EXECUTE flag – it can enable Permanent DEP. Once Permanent DEP has been enabled, it cannot be disabled anymore. The only way to enforce Permanent DEP is to make the application (like calc.exe) call SetProcessDEPPolicy with argument 1 - something you can do with shellcode.

Shellcode to enable Permanent DEP is rather simple: it only has to call SetProcessDEPPolicy with argument 1:

```
; Enable permanent DEP in current process
push PROCESS_DEP_ENABLE
call [KERNEL32_SETPROCESSDEPPOLICY-geteip+ebx]
```

When you inject this shellcode in your application, Permanent DEP will be turned on. But how can you modify your application so that it calls SetProcessDEPPolicy each time it is launched? You can inject the shellcode permanently in the application with a PE-file editor such as LordPE.

First you make a copy of the application (e.g. calc.exe) and you open it with LordPE. Then you create a new section with the shellcode, and make the entrypoint point to the shellcode. When finished, the shellcode jumps to the original entrypoint. You rebuild the PE file and save it.

When you execute this copy of calc.exe, your shellcode will be the first thing to run. This shellcode will enable Permanent DEP, and then jump to the start of the calculator program.

Example 3: Patching an application

Patches are changes to the binary code of an application. They typically fix bugs, security vulnerabilities or change features. But when you make changes to the files of an application (.exe or .dll), you invalidate the digital signature and you are probably breaking the EULA.

If you want to change an application but are not in a position to change the binary files, shellcode designed to patch in memory can help you.

Two years ago I developed a patch to fix an annoying “feature” of Adobe Reader 9.1. If you disabled JavaScript in Adobe Reader, each time you opened a PDF document with embedded JavaScript, Adobe Reader would remind you that JavaScript is disabled and propose you to turn in on again. To get rid of this nag screen, I developed a patch: replace byte sequence 50A16CBF9323FF90C805000039750859 with 50A16CBF9323B80200000039039750859 in file EScript.api. If you cannot change file EScript.api, you can still change the code directly in memory.

I have developed shellcode to search and replace a sequence of bytes in the virtual memory of an application. This shellcode can be used to apply the Adobe Reader patch I described. To achieve this, you inject this shellcode (together with the search and replace byte sequences) in Adobe Reader.

Another advantage of patching dynamically in memory with shellcode, is that the patch will not be lost when you update your application to a new version (Adobe Reader in our example).

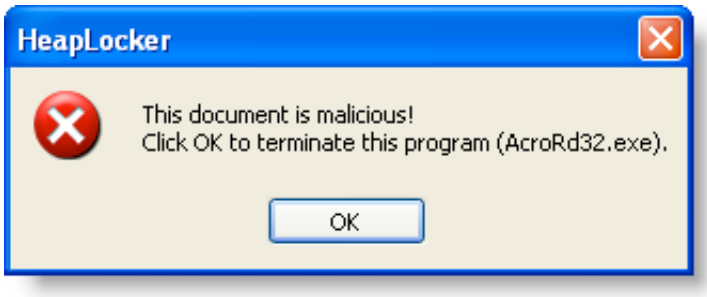
Example 4: Preventing heap sprays with shellcode

Shellcode is often used in attacks and malware together with heap sprays: the heap is filled with shellcode (preceded by a long NOP sled), and then the vulnerability is triggered. EIP jumps to the heap, hits a NOP sled and slides to the shellcode. The shellcode executes, and typically downloads and installs a trojan.

Successful heap sprays can be prevented by pre-allocating memory, so that the heap spray cannot write shellcode to the pre-allocated memory. If we pre-allocate memory and fill it with our own NOP sled and shellcode, we can intercept the attack and block it.

If you open a PDF document with an util.printf exploit with Adobe Reader 8.1.2, it will crash because this PDF document contains an exploit that makes EIP jump to 0x30303030 (this might be a few bytes off). Since there is no code at this address, an exception is generated.

But when we inject our own NOP sled and shellcode at this address (0x30303030), we achieve code execution. The exploit triggers, but it executes our shellcode, not the shellcode of the attacker.



This is because we planted our shellcode in the application's memory before the PDF document was opened and the heap spray executed.

The heap spray will fill memory with its attack shellcode, but it cannot overwrite our defense shellcode. So when the exploit triggers after the heap spray filled memory, our shellcode executes instead of the attacker's shellcode.

We could also use shellcode that suspends the attacked application and warns the user. For user applications, like Adobe Reader, this shellcode offers a huge advantage over protection methods that just pre-allocate heap memory and do not inject defensive shellcode.

If you just pre-allocate heap memory, the application will just crash when it is exploited, and the user will not know what happened. He could easily assume that Adobe reader just crashed because of a bug, and try to open the malicious PDF document again. Or even worse, send it to a colleague so that she can try to open the malicious document.

But if we inject defensive shellcode that displays a warning for the user, the user will know he is being attacked with a malicious PDF document and he will have a chance to act appropriately.

Conclusion

Shellcode is just a program, and it is up to the programmer to code the behavior of his program.

Shellcode is often programmed to attack, but there is no inherent reason why it cannot be coded to defend.

I hope that these four examples give you an idea how to use shellcode to protect your system. If you want the shellcode of these examples so that you can test it out yourself, take a look at my workshop exercises: workshop-shellcode.didierstevens.com

It also contains some tools (for example to inject shellcode), and I have produced a video for the DEP exercise.

Didier Stevens (Microsoft MVP Consumer Security, CISSP, GSSP-C, MCSD .NET, MCITP, MCSE/Security, RHCT, CCNA Security, OSWP) is an IT Security Consultant currently working at a large Belgian financial corporation.

He is employed by Contraste Europe NV, an IT Consulting Services company (www.contraste.com). You can find his open source security tools on his IT security related blog at blog.DidierStevens.com.

RSA Conference 2012

www.rsaconference.com/events/2012/usa

Moscone Center, San Francisco

27 February-2 March 2012.

The Amphion Forum

www.amphionforum.com

Hotel Bayerischer Hof, Munich, Germany

28 March 2012.

InfoSec World Conference & Expo 2012

www.misti.com/infosecworld

Disney's Contemporary Resort, Orlando

2-4 April 2012.

Cyber Defence Summit

www.cyberdefencesummit.com

Grand Hyatt Hotel, Muscat, Oman

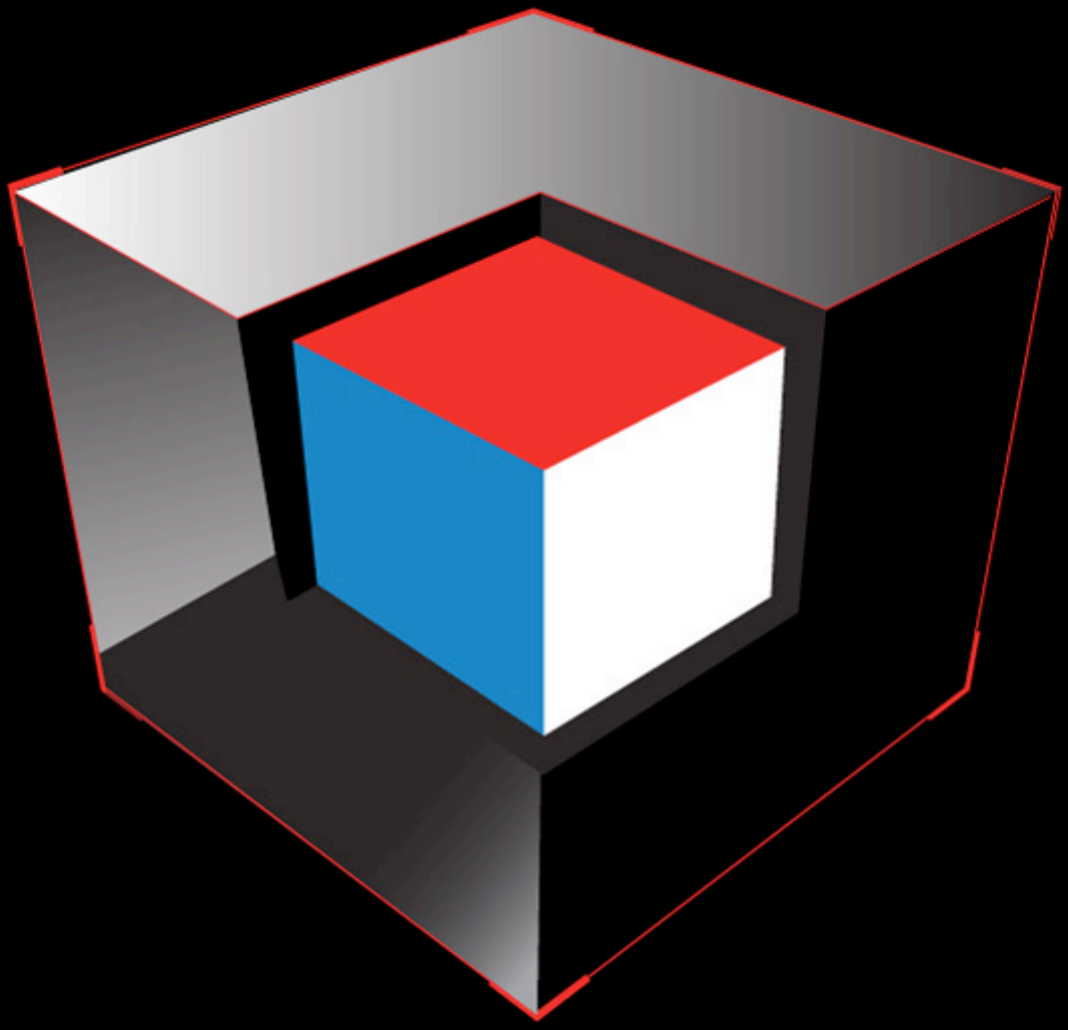
2-3 April 2012

HITBSecConf Amsterdam 2012

conference.hitb.org

Okura Hotel, Amsterdam, the Netherlands

24-26 April 2012.



hitbseccon2012

AMSTERDAM

May 21st - 25th @ Okura Hotel Amsterdam

REGISTER ONLINE

http://conference.hitb.org/hitbsecconf2012ams/

The Third Annual HITB Security Conference in The Netherlands featuring keynote speakers:

Bruce Schneier (Chief Security Technology Officer, BT)
Andy Ellis (Chief Security Officer, Akamai)

Using mobile device management for risk mitigation in a heterogeneous environment

by Keith Olsen and Elvis Gregov

Like it or not, enterprise IT organizations are quickly realizing that mobile devices are eclipsing PCs and laptops as the devices of choice for employees in the workplace and beyond. Mobile devices such as smartphones and tablets offer incredible power and flexibility in both our business and personal lives, which is leading to great pressure to integrate them within the enterprise.

Mobile computing today, when done right, creates an opportunity for workers to be more productive and happy, while also offering a major competitive advantage for the organization. However, if not done right, the consequences can be quite devastating.

This was the main topic of conversation during a recent series of workshops we hosted for public and private companies on the impact the proliferation of mobile devices is having on enterprises. Interestingly, not a single organization in attendance had a fully formulated Mobile Device Management strategy.

Most, if not all, were still on the ground floor trying to figure out what to do. They realize there are significant risk mitigation issues that they need to address, but because IT is often resource-constrained – especially in today's tight economic conditions – they continue to struggle to address these issues.

That is why many large, medium and even small corporations are seriously considering a formalized enterprise Mobile Device Management (MDM) strategy to deal with the proliferation of mobile devices knocking on their doors. This means not only using MDM specific applications and products, but also combining them with the right mix of policy, procedures and end user training.

Done correctly, enterprise MDM can be a practical approach that first assesses the organization's challenges, and then evolves with the dynamic, constantly changing business needs. By working together and developing a pragmatic approach with MDM, an organization's IT and business leaders are much more

likely to embrace today's mobile world – and benefit from it.

The mobility gold rush

It's not hard to see why these devices have spurred this gold rush to mobility in the enterprise. Sometimes, it comes from the top. The board or C-level execs may favor a certain device. Meanwhile, employees down the chain are often adopting the latest devices, platforms and applications much faster than corporate IT departments can react.

Social media is growing as a business application as well, blurring the work and home environments. Shifting business models also require tech-savvy employees, who are looking to connect to the enterprise with their iPhones, iPads, Androids, Blackberries and other mobile platforms. And along the way, employee expectations of corporate IT's ability to manage their mobile needs are changing.

But this consumerization of IT also presents some significant challenges. Of course, the cost of keeping up with the mobile world is always a factor. Many companies simply cannot afford to dedicate in-house resources to keep up.

Regardless of whether they do it themselves or engage outside expertise, organizations have to address the issue of integrating mobile into existing business processes. This includes managing the productivity of a remote workforce, determining the reliability of the mobile technologies, and most critical, security issues.

For instance, a recent joint study by Carnegie Mellon's CyLab and McAfee found that almost half of users keep sensitive data on their mobile devices, including passwords, PIN codes and credit card details. The ramifications of losing a device or having it compromised can be devastating – not only to the individual, but to the organization whose sensitive data, or at least the keys to it (passwords, PINs, etc.), may be held within the device.

For corporate IT, there are five major security risks that must be addressed:

- Physical access
- Malicious code
- Device and application attacks
- The interception of communications
- Insider threats.

Too often, the decision makers jump right to which tools they should buy and want to know what kinds of bells and whistles are out there to "lock these things down." To paraphrase former U.S. Secretary of Defense Donald Rumsfeld, when it comes to mobility there are "known knowns," "known unknowns," and "unknown unknowns." And most organizations don't know what they don't know when they look at how they are going to mitigate risk in a mobile environment.

So where do we begin?

In our opinion, it is always best to use those tried and true methodologies, or best practices, that security professionals have been preaching for years.

An effective approach begins with a risk assessment that assesses, evaluates, manages and measures each of these security risks. It is also important that the enterprise IT department work with the business units to understand their mobile requirements.

Without a comprehensive risk assessment, the purchasing decision will more than likely not reflect the reality of what they are looking to protect.

Before moving forward, organizations need to be able to answer several key questions:

1. How many mobile devices are connected to our network?
2. How do we know how many mobile devices we have?
3. How are these devices connecting?
4. How often are these devices connecting?
5. What data and services are these devices accessing?
6. How many of these devices are managed?
7. How many comply with our corporate policies?
8. What would be the ramifications if any of these devices are compromised, lost or stolen?

The matrix

From here, a matrix of controls can be developed to help enhance the risk mitigation. For instance, organizations need to determine what technologies and practices need to be implemented to control different classes of information that mobile devices can access or store. They also need to think ahead and extend acceptable use policies to all current and future mobile devices. And all mobile device users must agree to company-defined processes and regulations before being granted access to corporate resources.

The next step is to design effective training and communication plans. Although the overwhelming majority of organizations have policies in place for mobile devices, fewer than one in three employees are aware of their company's mobile security policy.

Consider this: many legit iPhone and iPad apps leak personal data to third parties. Users don't help – some still insist on using 0000 or 1234 as their password, making it easy to hack the device. Jailbreaking also puts iPhone users at risk for downloading infected applications, and also often leave the device with a standard root password that may grant an attacker administrator-level access to the device.

The threat is real. Just last year, a hacker pleaded guilty to electronically stealing data from more than 100,000 iPad users. Employees need to be aware that just because data is contained in electronic form on their phone, it is no less confidential and should be treated no less carefully than if it were on paper. And ideally, this requirement needs to be written into their employment contract and reinforced through regularly scheduled training.

One very simple, yet elegant, solution is to insist that users turn on the built-in security mechanisms on their devices. Even before establishing a thorough risk mitigation strategy, organizations can insist that users must install a PIN number on their iPhone if they plan to use it to access the network.

Mobile devices also have location awareness tools that can help the IT department conduct a remote wipe if the devices are lost or misplaced.

Although the overwhelming majority of organizations have policies in place for mobile devices, fewer than one in three employees are aware of their company's mobile security policy.

One size does not fit all

It is also important to realize that one size does not fit all when it comes to mobility. In fact, the ability to standardize on only one mobile operating platform within the enterprise is going the way of the rotary dial with the advent of these new devices and technologies.

Users are looking to blend their personal devices into their work lives, and that means organizations need to prioritize which devices they will support and at what levels. For instance, one issue that will need to be considered is what images will be displayed on the various operating systems. And security remains an ever-present concern, since nobody has yet been able to develop a universal centralized security app for the variety of phones being released by vendors to the market.

It is most likely that within any corporate environment there will never be a "one size fits all" solution. Employees, depending on their job requirements, will likely require varying levels of access to data and services. Thus it makes sense to consider some form of a multi-tiered answer to the problem. One suggestion is to segment the environment into three basic levels.

Tier One would be executives and others who need access to very specific types of highly sensitive information and services, and who will use the mobile devices as a critical facet of their jobs. Tier Two would be those whose mobile devices aren't a necessity for the corporation, but can benefit both themselves and

the organization with some access. Finally, Tier Three would be individuals to whom a minimal level of access (perhaps email only) is granted, but strictly as a convenience to the individual.

For this scenario a multi-tiered solution may look something like this:

• Tier One – Users qualify for corporate-liable devices and are provisioned with Mobile Device Management software and business applications.
• Tier Two – Users qualify for personally-owned devices that are "lightly" managed and supported by the organization.
• Tier Three – Users are free to connect their own devices with web-based applications, but they don't qualify for reimbursement of any kind, nor are they supported by the organization.

Organizations must also reserve the right to manage any and all mobile devices that require access to corporate resources. This management responsibility needs to be independent of who actually owns the mobile devices, and may require the installation of the firm's security policies on the mobile devices as a condition of being granted access to corporate resources.

One thing that can be easily overlooked is the need to protect the integrity and privacy of corporate data by isolating that data inside the firewall from personal data. This can be done either by "sandboxing" or taking a virtualized approach to data storage.

Of course, the key to this matrix of controls is enforcement of strong security policies that prevent data security breaches. These polices should address encryption, PINs and passwords, auto-lock capabilities, location tracking, remote wipes, disabling non-approved applications, features and functionality, and policy removal prevention.

Once all of these controls are in place, organizations can prioritize and determine how and when users will be provisioned with enterprise-class applications, and address ramifications for non-compliance with these controls. Enterprise MDM risk mitigation policies should also be reviewed at least yearly.

The apps story

Over 300,000 mobile applications have been developed in the last three years alone, and users have downloaded 10.9 billion apps over that same time period. Clearly, the proliferation of apps has helped drive the consumerization of IT.

The challenge is that most apps being published to the app store are developed autonomously and don't have a high level of quality assurance when it comes to security. Yes, Apple and others will say they provide security checks, but those are mostly rudimentary. Once the app is downloaded and installed, it is caveat emptor – back doors and coding objection flaws probably haven't been addressed in today's app stores. Users are at the mercy of the app, and they aren't really seeing what's being communicated and how it's being communicated across the network.

For instance, a colleague recently accessed a well-known airline's mobile app to check in. He was shocked when he immediately received a notification from his personal DLP (Data Loss Protection) service that his check-in request had been blocked due to a violation in the DLP security policy. It turns out that the airline's app did not enforce the transmission to be encrypted through a secure HTTPS connection, but rather simply passed it through clear text HTTP. So sensitive information – including his phone number, house address and flight information – would all have been transmitted had the DLP not stepped in and prevented it.

At the enterprise level, it's critical to understand which apps are mission-essential and standardize mobile users on those apps. Those can be published for download only while a user is on the corporate image and connected to the network. Organizations should also examine their internal app store and focus on setting restrictions on apps that are not business-essential.

Location. Location. Location.

The big problem with mobility is that organizations don't know where people are going to be when they try to access the network with their devices.

Whether they are sitting in a coffee shop or at a desk in their home or at work, users are more and more frequently looking to access their network through their mobile device than through a PC or desktop terminal.

So part of the risk assessment also needs to examine how users plan to connect to the network, where they will be using it, and what access points are acceptable. For instance, what will the corporate profile look like if a user is connecting through a hotspot at the airport, as compared to connecting via a wireless modem within the company's headquarters? It will also be important to decide how to authenticate to the access point itself. Will it be through a shared key, or will a third-part database be used to help authenticate users at corporate?

Organizations need to constantly keep their guard up when it comes to mobility. Employees will continue to adopt the latest devices, platforms and applications much faster than corporate IT departments can react.

However, by leveraging an effective security-centric approach to risk mitigation, organizations today can understand where the security risks lie, whether their operating systems are secure, if the mobile devices being used have adequate security features, and how to battle malware-laden code in applications. And that will let them – and their mobile users – rest easier.

Elvis Gregov and Keith Olsen are Solution Architects with Forsythe Technology Canada, Inc. (www.forsythe.com/na/aboutus/forsythecanada), an IT infrastructure integrator headquartered in Toronto, with offices in Edmonton, Vancouver, Winnipeg and Calgary.

HD is Chief Security Officer at Rapid7 and Chief Architect of Metasploit, the popular open-source penetration testing platform.

It's been a long road for Metasploit. What began as a personal project is now a major name in the security industry. How has the project evolved since it was acquired by Rapid7 and, overall, how has your professional life changed?

After almost nine years, Metasploit is still an incredibly fun project to work on. The acquisition by Rapid7, the development of commercial editions, and the dedicated development team have increased the project's capacity to grow and provide bigger and better things for security professionals.

The "corporate" environment and the expansion of our open source user base (nearly ~150,000 now) have not changed the soul of the project or the personal nature of contributions. Rapid7 is a strong supporter of open source, community collaboration, and just as importantly, common sense vulnerability disclosure. These traits are why Rapid7 was a good fit in 2009 and the driver behind our continued success, both commercially and as an open source project.

Metasploit has always been a platform for building security tools, testing out new ideas, and sharing those with a wider audience. Our recent move to GitHub and the increasing size of the community continues to prove that open collaboration is the best way to raise the bar within information security.

Through 2011, the project averaged more than one new module a day, with many of those coming straight from the community. A large portion of the team's time is spent working with contributors, tuning submitted code, and testing that code prior to rolling it into the master repository.

The Rapid7 team also handles things like quality assurance, core library changes, database architecture, and maintaining the build and installer environments for the open source code base.

The focus on community submissions has changed how we manage the project and where we allocate funds within the Rapid7 team. Our biggest revelation was that for some roles, we are better off focusing on the community submissions than trying to provide everything to everyone solely on our own.

This was an ego check in some ways, but it opened the door to faster progress and unexpected innovation. One example is the Railgun functionality within the Meterpreter extension.

Railgun provides a generic API for calling arbitrary Win32 methods and returning the results to the user. This code was dropped, anonymously, to the framework mailing list without a single follow-up from the original author. Since then, Railgun has been expanded, improved, and is now a driving force behind many of our post-exploitation modules and enables features that would have been time-prohibitive to do otherwise. This is a somewhat dated example, but it was a case of a one-off contribution changing the direction of the project in a way that we wouldn't have done on our own.

Staying involved with the security community is great for the open source project, but it also helps us align commercial development with the challenges our customers may yet to hit.

In many cases, new modules or features will land in the Metasploit Framework trunk, only to become critical features to our enterprise customers at a later time. A recent example includes a remote exploit for LifeSize video conferencing systems. On its own, this module did not appear to be that noteworthy - most of the Rapid7 team (not to mention our customers) had little experience with these systems and were not aware of their deployment scale or patch cycle.

STAYING INVOLVED WITH THE SECURITY COMMUNITY IS GREAT FOR THE OPEN SOURCE PROJECT

Fast forward two months to the introduction of the H.323 scanning module that was used to produce our recent "board room hacking" research, and the LifeSize module becomes immediately applicable. Not only does it increase awareness of "system" exploits for video conferencing equipment, but the H.323 survey results allowed us to see exactly how often these devices were patched and what percent of internet-exposed systems may be vulnerable.

The resulting news articles and blog posts resulted in many of our customers identifying these devices in their environment, proving that they were indeed a security risk using the LifeSize exploit module, and being granted the appropriate resources to fix the problem.

This was a great of example of a module submitted by another security company (SecureState) being combined with work from Rapid7 to identify and validate a real-world risk that many organizations had ignored to that point.

To this day, I stay actively involved in both the open source and commercial product development, as well as the media and social networking aspects of the project. In addition, I spend a lot of time on the phone with customers, handling support cases, working with integrators, and expanding the development team through new hires.

Having the perspective of a developer as well as sales, marketing, and support part of the business helps keep the project and our commercial products on the right path.

The biggest change has been handing off parts of the project to my co-workers and leaders in the community.

We have some amazing contributors (not to mention employees) and it has been liberating to share the load with this group of talented individuals focused on a common goal.

What major challenges did you face developing Metasploit on your own?

I was the founder of the project, but between 2003 and 2009 a handful of other developers were involved. In the early days, the core team consisted of myself, Matt Miller (skape), and spoonm. By 2007, I was the last man standing and this lead to the expansion of the team to include James Lee (egypt) and a number of new frequent contributors. Until recently, I was the only developer handling the release process, packaging, and testing, as well most of the project hosting, server administration, and legal paperwork.

Early on, Metasploit faced two major non-technical challenges.

The first was convincing the security community to use a brand new toolkit that aimed to replace one-off tools and exploits that had become second nature. We managed to solve this through brute force development and time. After years of off-and-on ridicule and frequent releases, many professionals finally tried the software and understood the point of our efforts. It took a couple more years before the work going into Metasploit was recognized as driving innovation, not just a replacement for existing solutions. Perseverance and continuous improvement was the only way to win this battle.

The second challenge was convincing the rest of the world that Metasploit was not designed to help under-skilled hackers break into corporate environments. The project was started at a time when vulnerability disclosure and exploit release was under attack by government bodies, security professionals, and the anti-disclosure underground.

Each of these groups had their reasons for opposing an increase in open security information, and the most telling trait was their shared opposition to projects like Metasploit. Over the next 5 years, I spent a lot of time defending the policies of the project, risking my own livelihood, responding to complaints, and generally fighting back against the perception that exploit tools did not improve security.

In 2008, the tide had finally turned. The "chilling" of open research and exploit development in the early aughts had contributed to a commercial environment where exploits were not just desired, but actually valuable. The huge (at the time, this meant ~150) number of exploits within the Metasploit Framework and the permissive BSD license meant that many organizations took a second look at the project and started using the framework for both internal and external work.

By the beginning of 2009, it was hard to find any organization providing network security services that was not a Metasploit user in some form. The day I realized we had crossed the tipping point was when I witnessed a sales associate demonstrating an IPS product using the Metasploit Framework on the exposition floor of the RSA conference.

Throughout all of this, the community continued to expand. Releases went from taking just weekends to almost an entire week of free time. Myself as well as the other project members all had full-time jobs, many of them within startup companies that demanded long hours as well. The discussion with Rapid7 and the eventual acquisition could not have come at a better time for the project or the open source community.

What advice can you offer to other open source security software developers?

There a few things I recommend:

1. Assign developer copyrights to a legal entity, such as a LLC or other limited partnership. This makes tracking expenses, registering domains, filing for trademarks, and handling copyright violations and other nastiness much simpler. In the case of the Metasploit Framework, the core developers assigned their rights to an LLC, which in turn provided the same developers with an unlimited license to use and repurpose the codebase.

2. Choose an open source license based on the goals you are trying to solve, not based on peer pressure or unjustified paranoia about corporate "abuse". At the end of the day, if someone wants to steal your code, they will, and in some cases just translate it to another language to build a competing product. This happens, so plan for it, stick by your goals and use copyrights, trademarks, and other legal

mechanisms to protect your brand where necessary.

3. Choose an open source license that will not cause a mountain of future work if you decide to change it later or build a commercial product. BSD-style licenses are a great choice. If you go with a license like GPLv2 or GPLv3, you may need to get copyright assignments from every single individual who contributed code to the project to effectively use that code within a commercial product.

The viral nature of GPL can seem like a great defense to commercial abuse, but that sword cuts both ways, and it can easily hobble your future efforts. This isn't to say that there is anything inherently wrong with GPL, just that it is not always the best default, especially for new projects. If you are intensely concerned with companies using your code without authorization, your best bet is to hire a lawyer to draft a commercial-style End User License Agreement.

This is definitely not open source, but it can buy time while you work out exactly how you want to license the project going forward. Two versions of the Metasploit Framework were released under a EULA-style license (v3.0 and v3.1) before the project converted back to a permissive BSD model.

CONTRIBUTORS WHO SUBMIT CODE YOU DON'T LIKE TODAY, CAN GROW INTO CORE DEVELOPERS OVER TIME

4. Identify a small number of goals that set your project apart from what is already available. Stay laser-focused on those goals until your project is the best fit for solving that type of problem. If you decide to expand the scope of the project, do so intentionally and commit to continuing in that direction for some time.

Users don't like features that work poorly and they definitely don't like to see features disappear due to a change of mind later on. Keeping focused on differentiators also helps drive awareness of the project and keep it top-of-mind for anyone trying to solve this problem.

If you are interested in merging your code into a larger project or in being acquired by a commercial entity, doing one thing really well makes the economics simple. You contribute a working implementation and deep knowledge of one area that would be more expensive for them to do on their own.

5. Stay friendly, stay humble, and appreciate the work being done by contributors, even if the code itself makes you gag. There are a number of security projects where the most common response to "do you know about X?" is a statement about how awesome one of their developers is and a story of how they helped them figure something out. Contributors who submit code you don't like today, can grow into core developers over time. Over the history of the Metasploit project, nearly all the major contributors started off with a couple patches and a longer discussion about coding guidelines and design goals. If you can find the time, make public your design philosophy and formatting standards.

As the project grows, document the process for contributing and set expectations about how long it takes to respond to submissions. Stay involved with the communities that use your project and recognize contributors who submit code, ideas, or just help answer questions for other users. In the open source world, the only real form of compensation is recognition.

Based on the feedback you get from your extensive user base, what are the most requested Metasploit features yet to be implemented?

To go by sheer volume, the two most requested features are the magic "hack everything" and "evade my antivirus" commands. A lot of time is spent in the community - whether its Twitter, IRC, mailing lists, or discussion forums - setting expectations for what the project can do and what the scope of our development is.

There are tons of great ideas submitted by the community, but we have to stay focused on what we do best (providing a platform for security tools and exploits) to continue pushing the project forward. In the case of a request not matching our acceptance guidelines (mass-automation modules, modules that don't meet our API requirements, etc.) we recommend that users simply fork the public repository and maintain it as a separate branch.

Most of the common requests boil down to current design limitations (more consistency between session types) or automating a chain of actions that would better fit into a plugin, a resource script, or outside of the core framework. We still see a lot of requests for additional payload capabilities, whether its new APIs for Meterpreter, expanded platform support, or stealth and evasion features. Strangely enough, what we don't see that often are requests for additional exploits.

METASPLOIT HAS EVOLVED FROM AN OPEN SOURCE FRAMEWORK THAT FOCUSED ALMOST ENTIRELY ON EXPLOITS TO A GENERAL-PURPOSE SECURITY PLATFORM WITH A MULTITUDE OF OPEN SOURCE AND COMMERCIAL OPTIONS. THIS TREND WILL CONTINUE

What is your vision for Metasploit in the next five years?

Metasploit has evolved from an open source framework that focused almost entirely on exploits to a general-purpose security platform with a multitude of open source and commercial options. This trend will continue. As security testing continues to move away from traditional exploitation methods, the open source core will evolve to support additional types of attacks, sessions, and data management.

The networking layer will continue to expand to support even more protocols and evasion methods. As much as we would like to avoid it, baking in additional payload-level evasion, specifically anti-virus systems, will become even more critical as user-assisted code execution becomes the predominant vector for remote exploitation.

Over the last two years, the database backend has gone through a number of major changes, and now ships enabled by default. This provides a new level of data persistence, storage, and automation capabilities through the PostgreSQL backend. Database architecture and data management will continue to play an important role in the design and functionality of the framework.

Scalability is another area where we have already made major improvements, but will need to continue growing to support the ever-increasing network sizes and exploits. The soft limit for concurrent sessions (open connections to compromised systems) is a bit over 1,000 today, per process, and this will likely need to increase.

One area that we have touched on, but not really dove into, is wireless protocol testing. Metasploit includes a number of modules for 802.11-based vulnerabilities as well as DECT station scanning and call monitoring, but the expansion of WiMax and new RF protocols will require new security tools to adequately assess their deployment. Metasploit may be the right tool for the job and we will focus development efforts accordingly.

In the end, it really depends on where the biggest risks are and what our open source users and commercial customers need to be successful. Metasploit has proven to be incredibly adaptive over the years, supporting everything from remote kernel exploits to serial-based wardialing and VoIP audio codecs.

Metasploit encompasses the open source framework, the free Community Edition of our commercial platform, and the flagship product, Metasploit Pro. We would love to apply the same modular automation and chaining techniques used in penetration testing to other areas of IT and Operations. How we get there will depend a lot on where we can help and whether we have the right capabilities to solve the problem at hand.

TARGETED ATTACKS WILL ALWAYS WORK FOR THE SAME REASON THAT CON ARTISTS STILL SUCCEED AT STEALING MONEY

We see the most dangerous elements in the threat landscape moving toward highly targeted attacks. What type of long-term impact will this have on the security tools we use today? Are we looking at a stronger artificial intelligence (AI) component in future computer security products?

The more things change, the more they stay the same. Prior to the glut of buffer overflow and memory corruption vulnerabilities, hackers still hacked, and most of this was focused on design flaws, logic issues, weak credentials, and exploiting the human behind the terminal on the other side.

Targeted attacks will always work for the same reason that con artists still succeed at stealing money. New protocols still ship with incredibly poor security measures and even mature technology introduces new flaws in the form of features.

A great example of this is the recent 802.11 WPS flaw (WiFi Protected Setup). WiFi security had finally reached the point where WPA2 with a strong password was good enough for many organizations. The introduction of WPS as a simple, secure way to access a network backfired by exposing millions of routers that would have been just fine using WPA2 alone. As technology continues forward, the folks designing new protocols and products will make mistakes, and just like before these will introduce introduce security flaws that can and will be exploited by malicious intruders.

Metasploit will stand ready to help our users and customers identify these risks and demonstrate their impact.

In August 2011, Rapid7 committed $100,000 to open source projects. How did this idea come about? What projects were chosen and how are they developing?

The Magnificent7 project is an idea that rose from the Rapid7 executive team as a straightforward way to contribute back to open source while driving progress in the area of information security.

This $100,000 budget will be split across 7 projects over the course 2012, focusing on specific milestones that the project creators identified as being the biggest roadblocks to reaching their goals. The first round of projects will be announced at the RSA 2012 conference in San Francisco.

Mirko Zorz is the Editor in Chief of (IN)SECURE Magazine and Help Net Security.

Defensive search-and-destroy "virus" delivered to Japanese government

It took three years and 178.5 million yen (around $2.3 m) to develop a defensive cyber weapon that can track down the sources of cyber attacks and disable them, but Fujitsu apparently did it.

Contracted in 2008 by the Japanese Defense Ministry's Technical Research and Development Institute, the company was charged with producing the aforementioned computer "virus" and a separate system capable of monitoring and analyzing cyber attacks.

According to The Daily Yomiuri, the virus is not only particularly effective when it comes to identifying the computers participating in DDoS attacks, but also the computers that control these botnets.

Unfortunately, it is supposedly less effective when identifying sources of attacks aimed at stealing information from targeted systems.

Upon delivery, the cyber weapon and the monitoring system were tested by the ministry in a "closed network environment", and have obviously proven to have been worth the money invested in them.

Citing client confidentiality as the reason, Fujitsu had so far declined to comment on the program. But even if the "virus" is as good as it seems, the question about whether it can be freely used by anyone is open for debate as the Japanese Parliament has recently made malware production and distribution a criminal offense.

Recycled cybercrime tactics adapted to conceal fraud

GFI Software released its VIPRE Report for December 2011, a collection of the most prevalent threat detections encountered during the month.

Phishing campaigns once again proved to be among the most significant threats, with scammers targeting Chase and Barclays customers, as well as launching malware attacks against Amazon shoppers expecting holiday packages.

“The threats we uncovered last month illustrate the consistent reuse of tried-and-true attack methods slightly modified to target new groups of potential victims,” said Christopher Boyd, senior threat researcher at GFI Software. “Most cyber-attacks at any given time rely on old techniques deployed with a new disguise. The reason we see them again and again is quite simply because they work, and we anticipate 2012 to bring many fresh takes on old scams.”

In a continuing trend highlighted in the last VIPRE Report, bank related phishing is increasingly becoming a common threat. Barclays customers received messages from a free Yahoo email address claiming that their account had been suspended due to incorrect login attempts.

The phishers employed scare tactics by insisting information had to be provided to reactivate the account within a certain amount of time. Once the victim’s identity was submitted, they were redirected to the official Barclays website in order to further mask the crime. Chase clients were targeted by a similar phishing campaign last month as well.

Another familiar cybercrime tactic that continued to gain momentum in December was scareware—fake antivirus software and system utility programs—that warn infected users of completely false threats to their computers.

The anatomy of the Gameover Zeus variant

The “Gameover” malware is a relatively new, "private" version of ZeuS. Support for the distributed command and control (C2) tools, integrated into the ZeuS botnet, were implemented at the request of one of the "private" clients of the ZeuS author.

Distributed C2 is a feature which was originally considered by the malware author in the ZeuS 1.4/2.0 beta program, but it was dropped from the final 2.0.x release because lack of demand among ZeuS customers in the face of significant coding and testing time. It was put back in as a feature during the recent, ongoing 2.2/3.0 beta program.

The “Gameover” version of Zeus also supports the use of complex web injections that allow the attacker to perform Man-in-the-Browser (MITB) attacks to bypass multi-factor authentication mechanisms. The ZeuS author has also rolled a Distributed Denial of Service (DDoS) component into the Gameover bundle.

Gameover has been used in this way. First, financial institutions were targeted with DDoS attacks against their online banking websites. These attacks were timed to coincide shortly after accounts at the targeted financial institution had fraud committed against them.

Chinese using malware to attack US DoD smart card security

AlienVault found evidence of Chinese-originated attacks against the US government agencies including the US DoD, which use a new strain of the Sykipot malware to compromise DoD smart cards.

One of the original versions of Sykipot was a Trojan horse application that opened a backdoor into the infected PCs. According to Jaime Blasco, AlienVault's Lab manager, this latest generation of diversified attacks may have been occurring as far back as March of last year, if not longer.

"This is the first report of Sykipot being used to compromise smart cards, and this latest version of the malware has been designed specifically to take advantage of smart card readers running ActivClient - the client application of ActivIdentity, whose smart cards are standardized at the DoD and a number of other US government agencies," he said.

"The smart cards are an important facet of security for the DoD – which manages the three main branches of the military in the US, the Departments of the Army, the Navy and the Air Force – and use the cards as a standard means of identifying active duty military staff, selected reserve personnel, civilian employees, and eligible contractor staff," he added.

So far, Blasco and his team have seen attacks that compromise smart card readers running Windows Native x509 software, which is reportedly in commonplace use amongst a number of US government and allied agencies.

This new strain, he says, is thought to have originated from the same Chinese authors that created a version of Sykipot late last year that piped out a variety of spammed messages with the lure of information on the next-generation unmanned 'drones' developed by the United States Air Force.

Identities of likely Koobface gang members revealed

First, details about a likely member of the "Ali Baba & 4" group (as they dubbed themselves) were made public by researcher Dancho Danchev and, as the story begun to unfold, security firm Sophos and the NYT revealed the names of the five individuals thought to be part of the Koobface gang.

Facebook started an investigation into the gang shortly after the Koobface worm first began to spread on the social network in 2008, and it took them only weeks to link the attacks to the suspects.

In 2009, independent researcher Jan Drömer mounted his own investigation. Starting with crucial information gleaned from one of the Koobface C&C servers and searching for links to it on the Internet - IP addresses, domain registration information, underground and legitimate forum posts, social network accounts and more - he made a beeline to the aforementioned group of individuals.

According to him, there is a variety of reasons behind the success of the Koobaface gang: they misused powerful online services to spread the worm, didn't overdo on the size of the botnet, haven't aimed at making the worm perfect but invested just enough revenue to earn more than enough money, and have operated in countries whose law enforcement agencies haven't a good record when it comes to cooperating with their US and European counterparts.

"Frankenmalware" active in the wild

If you're not careful and you don't use anti-malware software, you might end up with various viruses, Trojans and worms on your computer. But, according to Bitdefender researchers, you might even get saddled with a hybrid or two of this different types of malware.

The researchers have dubbed these hybrids "frankenmalware", and out of some 10 million detected and analyzed malicious files, they identified over 40,000 of these "malware sandwiches".

"A virus infects executable files; and a worm is an executable file," explained Loredana Botezatu. "If the virus reaches a PC already compromised by a worm, the virus will infect the exe files on that PC - including the worm. When the worm spreads, it will carry the virus with it. Although this happens unintentionally, the combined features from both pieces of malware will inflict a lot more damage than the creators of either piece of malware intended."

To explain how the symbiosis works, she shares the example of the Virtob virus/ Rimecud worm "collaboration".

The Rimecud worm spreads via file-sharing apps, USB devices, Microsoft MSN Messenger and locally mapped network drives. Besides that, it also steals passwords by injecting itself into the explorer.exe process, opens a backdoor that will allow it to download additional malware from a C&C server and - if the computer has remote control software installed - allows cyber criminals to access it and control it.

As it turns out, Bitdefender has recently begun spotting the Virtob virus attached to the aforementioned worm. The virus - which also opens a backdoor, contacts IRC C&C servers, modifies a host of files - infects executable files and, as the worm itself is an executable, it is also likely to be infected.

A peek into the Sykipot campaigns

Symantec researchers have recently discovered and managed to take a peek into a staging server for the Sykipot campaigns, which was also occasionally used as a C&C server for delivering instructions to the malware installed on the compromised computers.

In it they discovered many things that gave them insight into how the campaigns are differentiated and waged.

"Each campaign is marked with a unique identifier comprised of a few letters followed by a date hard-coded within the Sykipot Trojan itself. In some cases the keyword preceding the numbers is the sub-domain's folder name on the Web server being used," they shared. "These campaign markers allow the attackers to correlate different attacks on different organizations and industries."

The location of the server (Beijing), those of attackers contacting it (Zhejiang province) and Chinese words contained in path and some file names seem to validate the theory that Chinese hackers are behind the attacks.

The researchers found over a hundred of of malicious files sent as attachments to the targets. They were mostly specially crafted PDF files that would drop the Trojan onto the targeted system once they were run.

THE AMPHION FORUM2012

28 March 2012
Munich, Germany

Devices outnumber PCs on the Internet by five to one… and they're proliferating fast.

Still, >99% of connected devices have no ability to defend themselves against hackers or malware. Attacks on smartphones already net huge sums for Internet criminals – and devices used in critical contexts like medical implants and industrial automation are even less secure.

Let's do something about it.

Join hundreds of the brightest minds from business, academia, government and defence, as we reach for a common goal: a safer, more secure Internet of Things.

Participants Include:

accenture

intel

McAfee

Symantec.

Smartphone Security
Smartgrid Security
Medical Device Security
Defense Electronics Security
Consumer Electronics Security
Industrial Automation Security
Gaming Security
Datacom Appliance Security
Automotive & Aviation Security
Mobile App Security

www.amphionforum.com

Using and extending the Vega open source web security platform

by David Mirza

The last decade has seen a major shift in the notion of perimeter exposure. Firewalls are robust and modern operating systems are increasingly hardened by default.

The new perimeter is the web application, unseen by the firewall, universally exposed, rich with complex functionality, often consisting of a mix of custom and third party code. Shorter development cycles for web applications means that codebases change with a higher frequency. Meanwhile, developers are commonly inexperienced and lack tools to help. While many open source tools exist, most of them can be tricky to use by non-security professionals. These are some of the reasons that make managing the security of web applications a challenging problem.

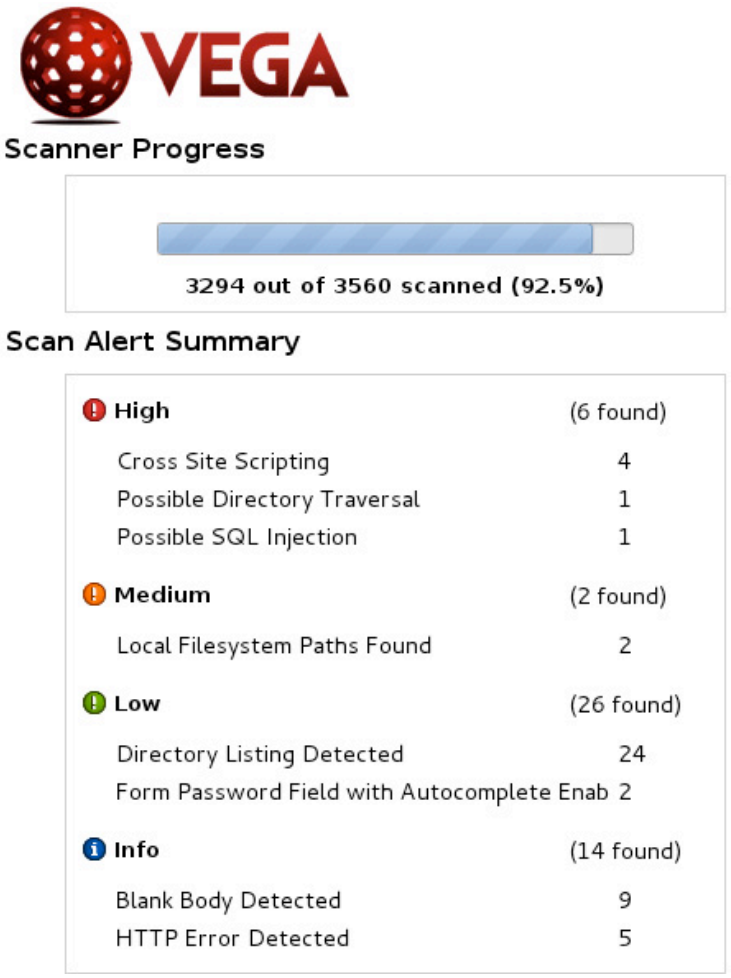
Vega is a new open source platform for testing the security of web applications developed by Subgraph (www.subgraph.com) and released under the Eclipse Public License (EPL) 1.0. Vega is written in Java, is GUI-based, and runs on OS X, Linux, and Windows.

The 1.0 beta was included in BT5R1 and later. Users interested in building bleeding-edge Vega from source can obtain the source code from our repository, hosted at github.com/subgraph/Vega. Vega can be compiled by simply running “ant” (note that the build script will download dependencies from a Subgraph server). To build the newest version of Vega:

```
$ git clone
git://github.com/subgraph/Vega.git
$ cd Vega
$ git checkout develop
$ ant
```

After a successful build, the binaries will be in:

```
$ ls build/stage/I.VegaBuild/
VegaBuild-linux.gtk.x86.zip
VegaBuild-macosx.cocoa.x86_64.zip
compilelogs/
VegaBuild-linux.gtk.x86_64.zip
VegaBuild-win32.win32.x86.zip
VegaBuild-macosx.cocoa.x86.zip
VegaBuild-win32.win32.x86_64.zip
```



Vega includes a crawler for automated vulnerability scanning, as well as an intercepting proxy for manual hacking. While Vega includes a set of built-in vulnerability checks, the real power of Vega comes from its extensibility: there is a built-in Javascript interpreter for creating custom modules using a rich API. In this article we will describe all of the features of Vega and walk through simple examples of custom module development for each of the two types.

Vega is based on Equinox OSGi and Eclipse RCP, the modular framework and UI toolkit underlying the Eclipse IDE. Vega also incorporates the Mozilla Rhino Javascript interpreter, Apache HC, jsoup and db4o. Development has continued since the 1.0 beta release on July 1, 2011. In this article we will demonstrate some new features in the pre-1.0 version available from our repository at github.

Basics

The two core modes of operation for Vega are as an automated scanner and as an intercepting proxy. The Vega user interface is split into two corresponding "perspectives" (arrangements) of UI components known as "views". This terminology may be familiar to users of the Eclipse IDE. In the current version of Vega, there is a scanner perspective and a proxy perspective. Views within each perspective can be moved around and re-sized. Selecting "reset perspective" from the Window pull-down menu in the Vega toolbar will reset the perspective to its default arrangement, should the user ever want to return to the initial configuration.

Vega saves scan/proxy data and configuration settings in a data store known as a "workspace". The workspace can be cleared by selecting "Reset Workspace" from the "File" toolbar menu. The workspace can be backed up or transferred by locating or moving the "model.db" file. On Linux systems, this file will be in a sub-directory within `~/.vega/workspaces`.

The scanner UI is the default perspective, presented when Vega is run for the first time. We will therefore describe the scanner first.

Scanner

The Vega automated scanner is a vulnerability assessment tool that crawls web applications, actively and passively probing for known and unknown vulnerabilities using customizable Javascript modules and Java probes.

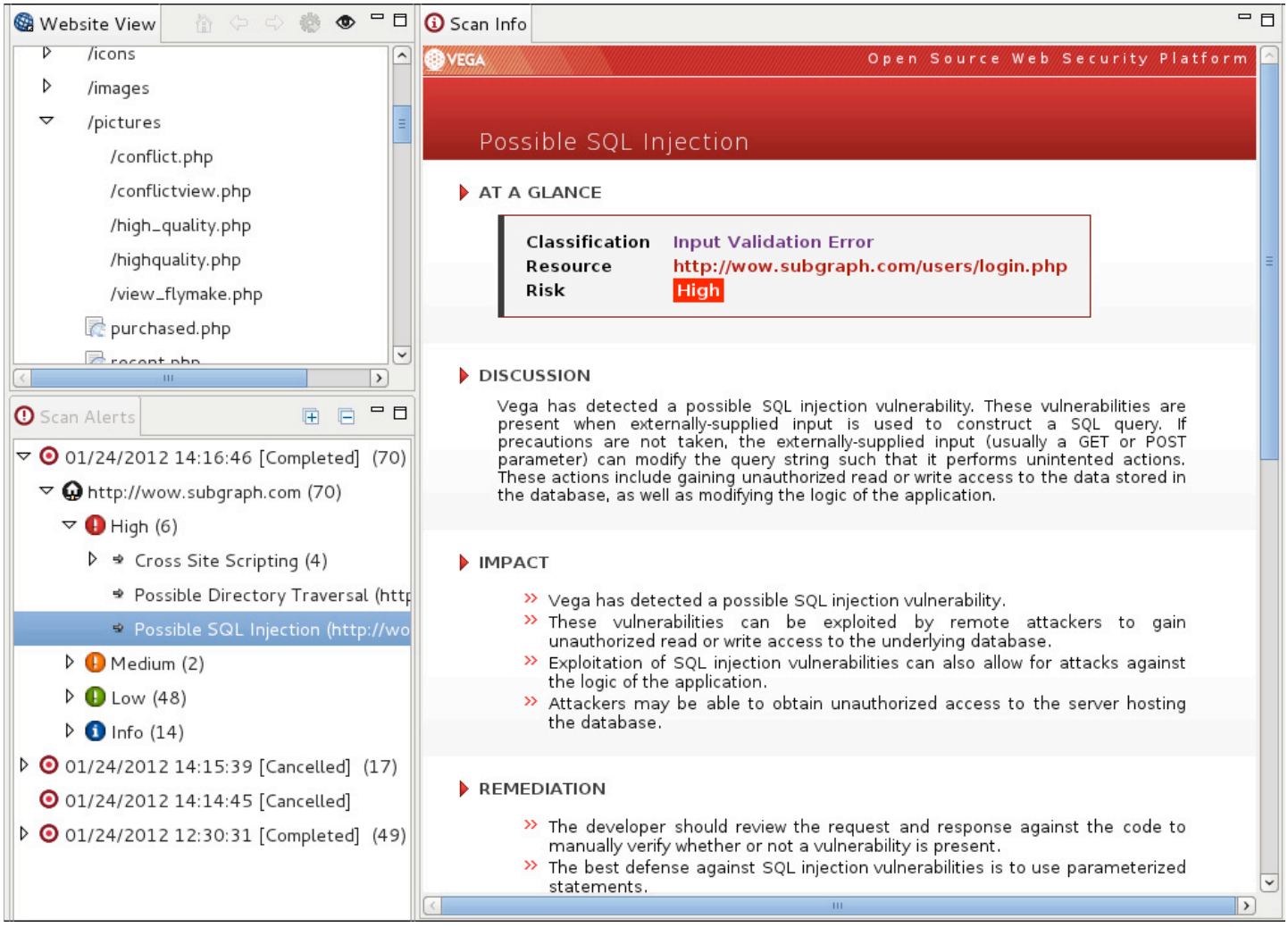
The scanner interface has four sections. In the top right is the website view, where a tree of web paths seen and visited by Vega will be rendered. The website view presents data in hierarchical order: for each website, the arrow icon to the left of the hostname can be clicked to expand it into a list of paths discovered on the server. Sites and paths that are grayed out indicate that Vega has seen but not accessed them. For example, this can occur when Vega crawls a website and discovers a link to a host or path outside of scan scope. There is a button above the web view to remove these unvisited paths from the list.

The user can also select a website or path and instruct Vega to begin scanning from that point.

Just below the website view is the Scan Alerts view. This area is where alerts generated by modules during scans or proxy usage will be listed. Each individual scan that has been run, known as a scan instance, will have its own tree of generated alerts listed in order of severity, grouped by type. The proxy has its own tree for alerts. Like modules, alerts are entirely customizable. Alerts rendered by Vega contain static content from XML template files and dynamic content from the modules that generate them. Users can edit existing alerts or create their own new ones very easily.

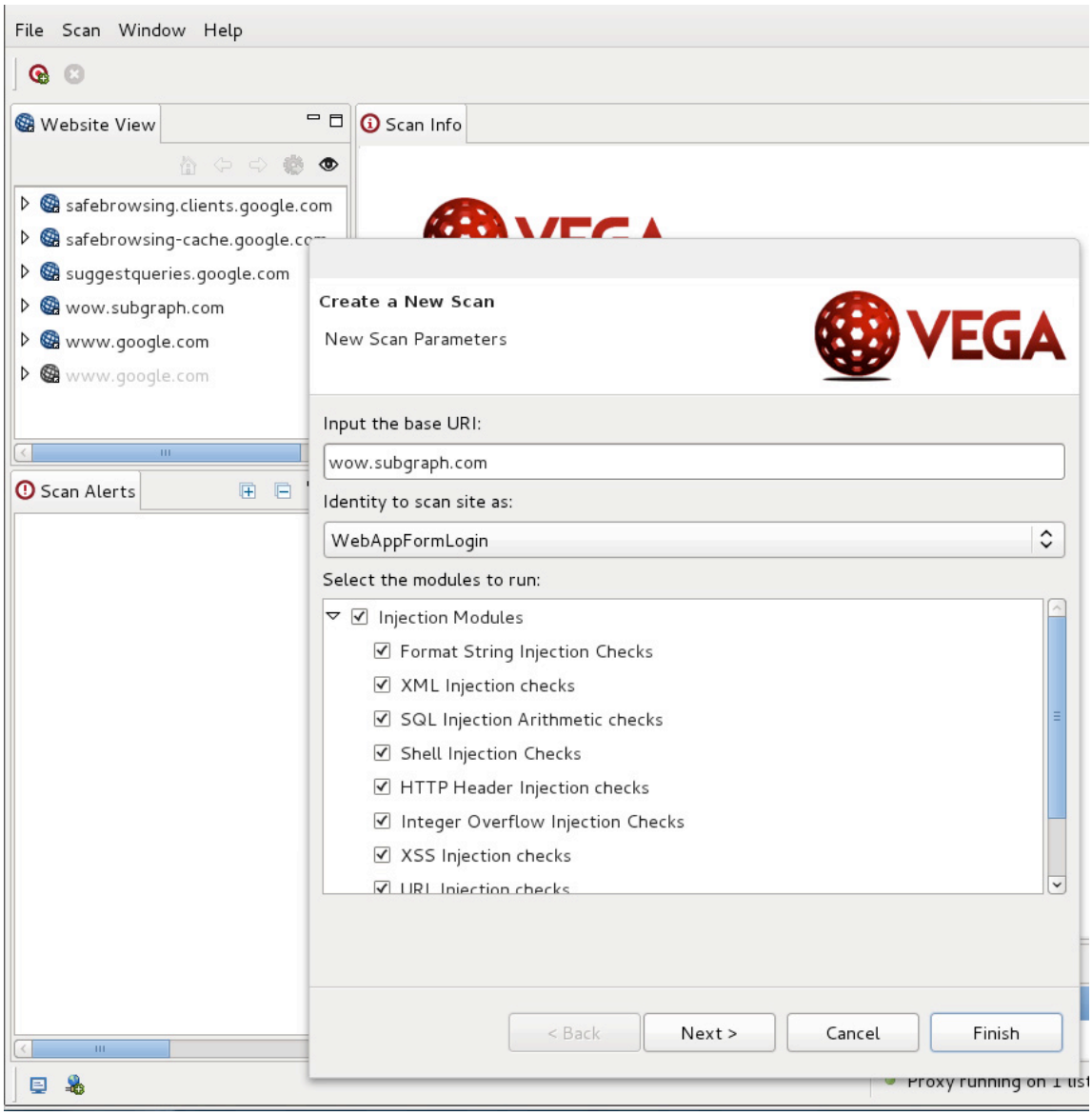
Vega has a general console for text output. This is where output is printed when the debug setting is enabled for the scanner. The modules also send their debug output to the console.

The console is accessible by clicking the console fastview icon in the bottom left corner of the Vega UI. The fastview icon will blink with a warning indicator when there is pending output.



In its most basic usage, the Vega automated scanner crawls a website, running vulnerability detection modules written in Javascript. To start such a scan, the user can click the "Start New Scan" target icon in the top right corner of the scanner perspective. Doing so will prompt a pop-up dialog with some scan parameters to be set by the user.

The base URI field is the starting point of the crawler: for example, the user could input www.example.com. The identities field is for assigning a set of credentials that Vega will use during the scan. This is used when the application being scanned requires authentication.

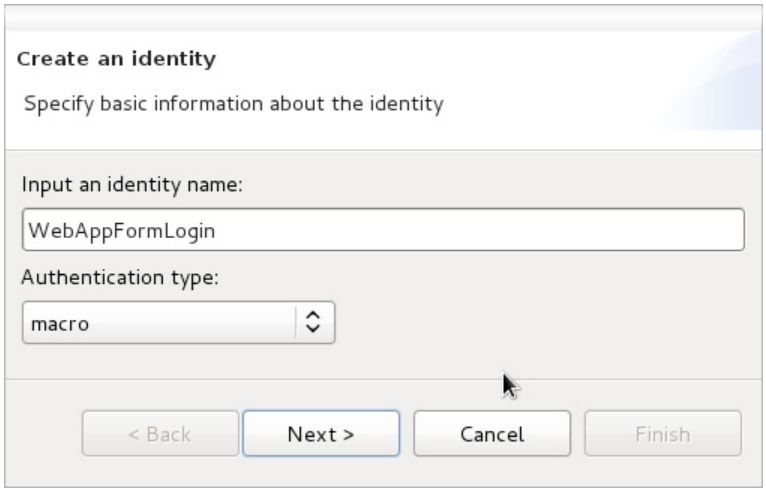


Identities

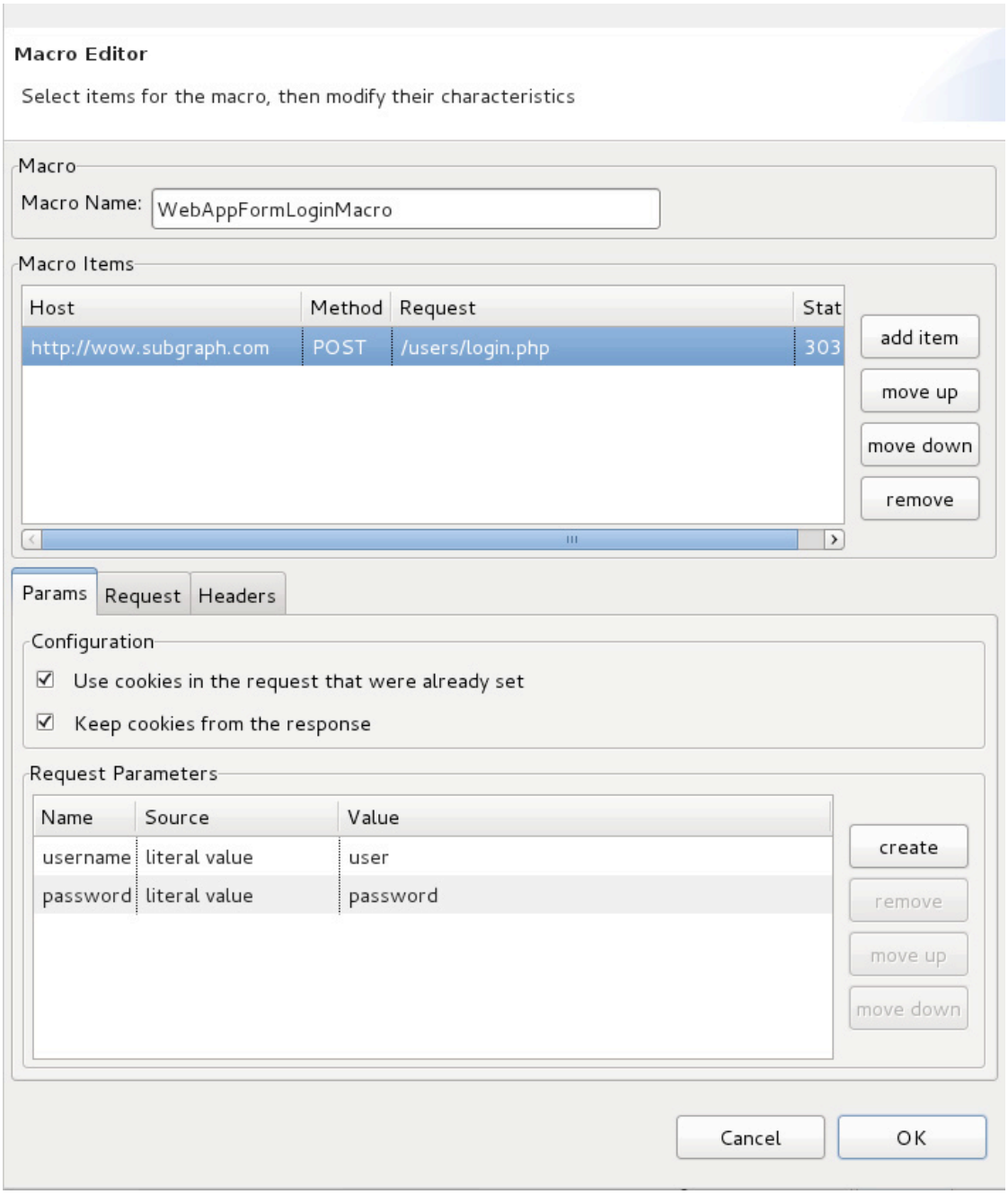
Identities are the general facility provided by Vega to store sets of credentials. Identities can be created for various authentication mechanisms, including basic, digest, and NTLM.
For form-based authentication, it is possible to bind an identity to a macro, which instructs Vega to authenticate using a recorded set of requests. Macros can be created before an identity is created, or during the process.

Macros

Vega allows for sequences of requests to be recorded and replayed before the start of a scan. These sequences are known as "macros". This function is useful for automatically replaying login form submissions to establish an authenticated session for the scanner. A macro can be created one of two ways: by clicking "Create Macro" button in the macro view, at the bottom of the scanner perspective, or from within the identity creation dialog.



To create a macro, the user should first perform the requests through the proxy with a HTTP client. These requests can then be selected from a request table of recorded proxy requests within the macro interface. By default, cookies are preserved. The user may also add or modify HTTP header fields in the macro requests. The macros are given names by the user and can be saved.



Below the identities selector is a tree of modules that can be selected or deselected for inclusion in the scan. Modules typically represent individual vulnerability checks and each module is a single Javascript file in the Vega modules directory (scripts/scanner/modules/ injection or response). Vega supports two types of modules: "basic" (active) and "response processing" (passive).

The basic modules, which are also known as injection modules, run on each injection point identified by the crawler: all files, directories, and parameters. The basic modules do the fuzzing: they generate multiple new requests and process the responses using a callback function that they register. The response processing modules run on all HTTP responses received by Vega, grepping for patterns corresponding to security vulnerabilities. Both modules can generate alerts and store/ retrieve data in an internal database. Adding a module to the list is as easy as dropping a file in the right directory. Modules can also be edited and reloaded without restarting Vega.

To continue the setup of a new scan, the user can click "Next" to continue to the second step, or skip it. For the purpose of this tutorial we will proceed by clicking "Next". The second step allows for the user to input a custom cookie value, as well as any paths that they do not wish the crawler to access. This is useful if there are logout links in the application that will clear the authenticated session if they are accessed. Clicking "finish" will start the scan.

Once the scan has started, the progress is indicated in the central view of the scanner perspective. The progress bar will adjust in size as the scanner discovers more of the application during its recursive crawl. Vega performs various tests on each accessed path, trying to determine if it is a file or directory. Vega also does 404 analysis to fingerprint the server response in cases where a path that does not exist is accessed. As Vega identifies vulnerabilities, the summary table in the "Scan Info" central view will be populated and corresponding alerts will be added to the Alerts view. When an alert is selected for review from within the Alerts view, it will be rendered in the Scan Info central view. The alert contents will be described in more detail below.
Clicking on the top level node of the scan instance in the Alerts view will switch the contents of the central view back to the scan summary.

Scan alerts

Vega modules generate alerts when they detect possible vulnerabilities. It is up to the module developer to decide when and why to generate an alert, and which alert should be generated. The alerts are generated when the module invokes a specific method in the module's context, specifying the XML template to use for the alert. It also passes parameters such as a HTTP request and response, some relevant content, a link to the vulnerable resource on the target server, and a unique key for the alert to prevent duplicate instances. Vega assembles the final alert using static content from the XML template file and dynamic content from the module.

Example - Module "vinfo-1918.js" invoking an alert (the first parameter is the XML template):

```
[..]
    ctx.alert("vinfo-1918", request, response, {        // XML file, request
object, response object
      output: result.join(" "),                         // output included in
alert
      resource: request.requestLine.uri,                // vulnerable resource
(link)
      key: "vinfo-1918" + request.requestLine.uri + result.join(" ") // unique
key
    });
[..]

Example - XML template "vinfo-1918.xml":

<?xml version="1.0" encoding="UTF-8"?>
```

```
<alert>
<title>Internal Addresses Found</title>

        <class>Information</class>
        <severity>Low</severity>

        <impact>May reveal internal network structure to outside attackers.</
impact>
        <impact>Internal IP addresses that have been disclosed could be used
as targets in otherwise blind attacks.</impact>

        <discussion>
        Vega has discovered references to internal hosts or networks in
publicly accessible content. These addresses may reveal information to an
attacker about the internal network structure, increasing the likelihood of
success for blind attacks involving other vulnerabilities.
        </discussion>
[..]
```

These will be discussed in further detail later in this article.

Request viewer

Most Vega modules save a specific request and response pair for inclusion in an alert. This is useful for users who want to verify that the possible vulnerability is present or investigate it further. This request and response pair is made available to the user in a link within the alert. Clicking on the "Request" link will open up the request viewer fast view, with the request and response in message viewers below a request table. The user can then inspect the full HTTP request and response associated with the alert. Right clicking on the request log entry in the request table above the message viewers allows for it to be selected for replay.

If a request is selected for replay, a request editor tab will open in the Scan Info view. The user can then modify the request and click the "play" button above the editor region to transmit it to the server. The server response will be rendered in the message viewer below. The user may modify and send as many requests as they like from this view and then close it when they are finished.

Scanner preferences

The scanner preferences allow for resource limits to be set. This can constrain the scope of scans. One useful debugging feature during module development is the logging of all scanner requests (by default, only scanner requests saved by alerts are logged). The user is advised to review the settings in the preferences menu option of the "Window" menu bar.

Proxy

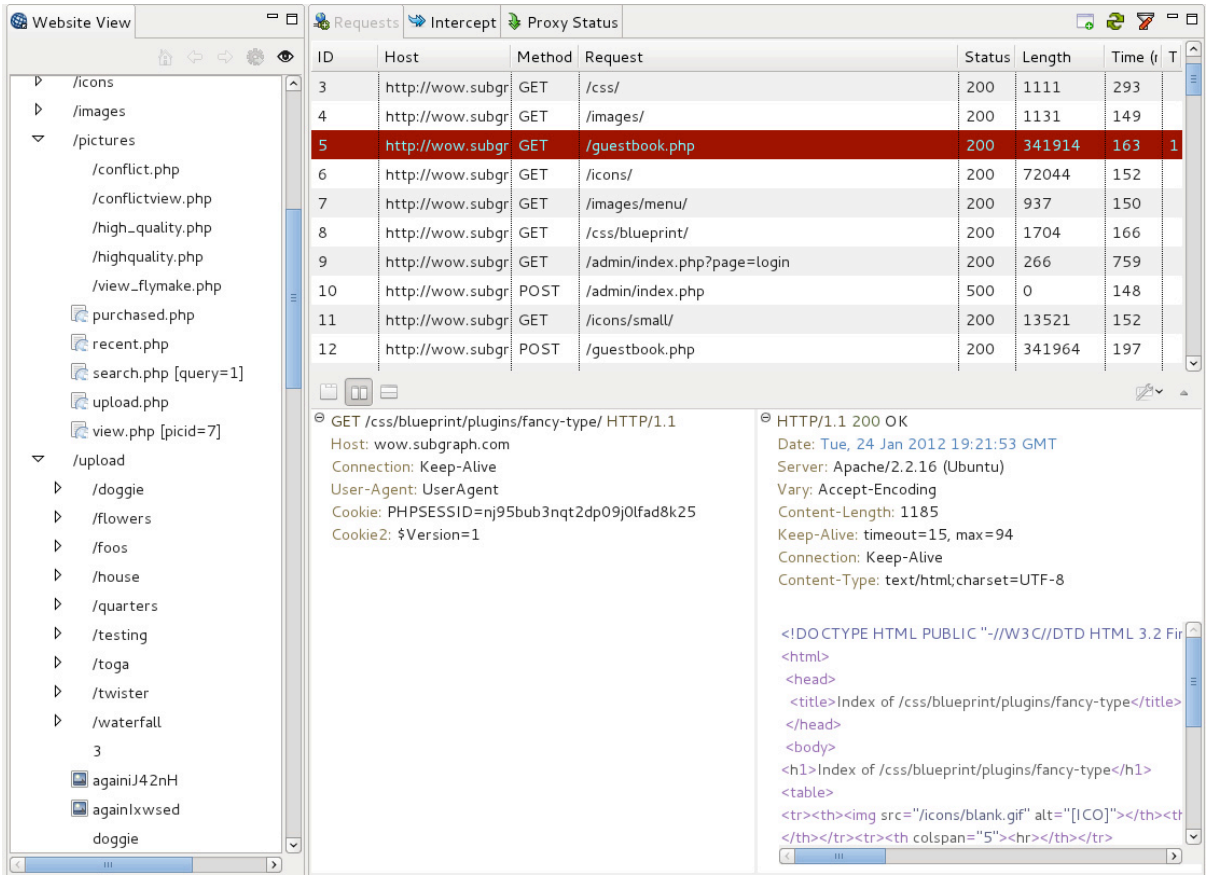
The Vega intercepting proxy is meant for use with a HTTP client, such as a web browser, and allows for close observation and manipulation of client-server interaction. When the proxy is enabled, Vega opens a listening TCP port on a configurable port number (default is 8888). HTTP clients can be configured to use the proxy on this port. Firefox is a good choice of browser for use with the proxy because it maintains its own proxy settings, distinct from system-wide proxy settings.

The Vega intercepting proxy can be accessed by clicking the proxy button at the top right, which will open the proxy perspective. The proxy can be enabled by clicking the "Play" button in the top left corner of the proxy interface, and can be stopped by clicking the stop icon. A status indicator in the bottom right corner of the Vega UI will indicate that the proxy is listening. The proxy perspective is comprised of three major views: the website view, the request table, and the HTTP message viewers. The website view in the proxy is identical to the small version embedded in the top left corner of the scanner perspective. The request table is a list of all requests saved by Vega.

Request table

By default, all requests and responses that pass through the proxy are stored in Vega's underlying database. The contents of this database can be viewed in request tables, an arbitrary number of which can be created, each with specific filters applied. The request list can be filtered by criteria such as regexp matching paths and status code. Clicking the "recycle" icon will reset the filter. If multiple filters are needed, it is possible to create additional request tables to which other filters can be applied by clicking on the "Open New Request Viewer" icon above the request list.

Right-clicking a row in the request list will bring up options such as replaying the request and tagging it. Requests can be tagged and assigned highlighting colors to distinguish them if they are of some specific interest. Clicking on replay request will open a request editor tag. The request can then be edited and re-transmitted an arbitrary number of times.



HTTP message viewer

HTTP requests and responses are rendered in a component called the message viewer. There is a message viewer for the request and the response. The arrangement of these viewers is configurable - while the default is tabbed (request, response), the positioning can be changed by selecting one of the icons above the message viewer views.

The message viewer is meant to serve as a container for rendering content in HTTP messages, including headers and message bodies. The message viewer supports rendering of some complex structured data, and will improve in future versions of Vega. Presently the Vega message viewer supports rendering of syntax-highlighted markup, binary image content, and binary data in hexadecimal representation. The Vega development team plans to improve substantially in this area, adding support for a variety of types of structured data. Within the message viewer are two sections: the HTTP headers and the message body. The header can be collapsed to make more room for the content. There is also an icon to hide the request table and fill the UI area normally occupied by both with the message viewer, creating more room for inspecting the content of a message pair.

Configuring interceptor rules

The Vega proxy can be configured to intercept HTTP requests and responses passing through it. When a message is intercepted, it is held by the proxy until the user chooses to drop it or forward it. Pending messages can be modified before they are forwarded. The interceptor can be set to intercept all messages, or only those that match certain criteria. Examples of criteria for interception include method type, status code, and regexp matching on hostname or path. For example, it is possible to configure an interception rule so that all outgoing requests for /vulnerable.php are intercepted, while all others are passed through.

An indicator at the bottom of the Vega interface will notify the user when an intercepted message is pending. Clicking the button will take the user to an interface where the pending request can be edited and then forwarded or dropped. When multiple requests are pending, it may be more useful to view all of them in a table. Clicking the "Proxy Status" tab brings up such a table. Multiple rows can be selected and forwarded or dropped at once.

SSL

For observing/manipulating communication between a HTTPS client and server, Vega performs a dynamic man-in-the-middle certificate injection when SSL is encountered. This can (and should) cause a certificate error in connecting HTTPS clients, as the certificate injected by Vega is not issued by a trusted CA. For convenience, it is possible to have Vega generate a CA certificate that can be imported into a client's certificate store. To generate this certificate, visit the magic proxy URL http://vega/ca.crt with a browser configured to use the Vega proxy. With Firefox, the user will be asked if they wish to import the certificate. The certificate may need to be saved and then manually imported into the certificate store for other HTTPS clients.

Response processing modules

It is possible to run response processing modules during use of the Vega proxy. Most of them are set to run by default. The tool icon to the right of the proxy stop icon brings up a list of the response processing modules selected for use with the proxy. Alerts triggered by these modules during proxy usage are listed in their section in the Alerts view in the scanner perspective.

Having explained the scanner and proxy, we will now walk-through extending Vega through the development of custom modules.

Extending Vega

Vega modules are written in Javascript and are available to use when placed in the correct directory - restarting Vega should not be necessary. Modules can also be modified without necessitating a restart. On Linux systems, this directory is in scripts/scanner/modules. There are two additional subdirectories, injection/ and response/, used for storing the two respective types of modules.

Response processing module

Response processing modules run on every response received by Vega. They process responses to scanner-issued requests as well as responses passing through the proxy. In this tutorial, we'll look at the vinfo-email.js module, located in scripts/scanner/modules/response.

The first requirement of any response processing module is a module object. This object supplies the name and category of the module to Vega. It can also supply a flag to indicate whether or not this module should be disabled by default.

```
var module = {
  name: "E-Mail Finder Module",
  type: "response-processor",
  defaultDisabled: false
};
```

Some of the modules are set to be disabled by default. This may be desirable for a variety of reasons, such as their computational cost or the number of false positives they produce.

The entry point of a response processing module is a function called run() that accepts three parameters:

```
function run(request, response, ctx)
```

These parameters are: an object representing the HTTP request, the HTTP response, and the context. The context object connects the module to Vega and exposes the scanner API to the module developer. These objects are documented in detail on the Subgraph documentation website: https://support.subgraph.com/trac/wiki/ResponseProcessingModules

Response processing modules are invoked when a response is processed. The logic of the module occurs within the run() function.

In this example the module is analyzing the body of the responses using Javascript regular expressions to try and identify email addresses:

```
function run(request, response, ctx) {
  var atDomainRegex = /@(?:[^\s.]{1,64}\.)+\S{2,6}/,
  mailRegex = /\w[^\s@]*@(?:[^\s.]{1,64}\.)+\S{2,6}/g,
  strictMailRegex =
/[\w!#$%&'*+-\/=?^`{|}~.]+@(?:(([a-z0-9]{1}[a-z0-9\-]{0,62}[a-z0-9]{1})|[a-z])
\.)+(?:aero|arpa|biz|com|coop|edu|gov|info|int|mil|museum|name|net|org|pro|tra
vel|mobi|asia|xxx|[a-z][a-z])/i,
  body = response.bodyAsString,
  emails = [],
  r, sr, i, found;

// First the module attempts to find a basic match of characters@domain, if it
does not, it returns

  if (!atDomainRegex.test(body)) return;

// It then attempts to match a more strict regular expression. Any matches are
converted to lowercase and uniquely stored in an array

  while (r = mailRegex.exec(body)) {
    sr = strictMailRegex.exec(r[0]);
    if (sr && emails.indexOf(sr[0]) == -1) {
      found = 0;
      for (i = 0; i < emails.length; i++) {
        if (emails[i] == sr[0].toLowerCase()) {
          found = 1;
        }
      }
      if (!found) {
        emails.push(sr[0].toLowerCase());
      }
    }
  }

The processing of the response is complete, and the alert can now be generated
if e-mail addresses were identified:

  if (emails.length) {
```

A unique key is then constructed for this alert. The key in this example is constructed by sorting all of the discovered e-mail addresses and delimiting them with a space. The key is arbitrary - it is up to the module developer to come up with a scheme that prevents too many duplicate alerts while still providing useful findings. In this case, the key should prevent other alerts from being generated for the same precise type of finding:

```
var key = emails.sort().join(" ");
```

```
var uristr = String(request.requestLine.uri);
    var uripart = uristr.replace(/\?.*/, "");

The alert() function is exposed through the context object:

    ctx.alert("vinfo-emails", request, response, {      // The XML file, the
request, and response objects
      "output": emails.join(" "), // The output to be rendered in the alert
      "resource": uripart,        // The URI for the resource field of the
alert
      key: "vinfo-emails" + uripart + key  // The unique key
    });

  }
```

Regular expressions are not the only tool that can be used to analyze response content. The Vega Javascript API comes with JQuery to analyze content at the DOM level. A DOM object can be obtained by accessing response.document. When the module does this, Vega will lazily attempt to parse a DOM from a response body. If it succeeds, a DOM object will be returned. Otherwise response.document will be null. JQuery can then be used on the DOM. An example of this module in use is in scripts/scanner/modules/response/vautocomplete.js:

```
[..]

  if (response.document) {
    var form = jQuery("form", response.document);
    form.children().each(function() {
      if ((this.getAttribute("type") != null) && (this.getAttribute("type") ==
"password")) {
        if ((this.getAttribute("autocomplete") == null) ||
(this.getAttribute("autocomplete").toLowerCase() != "off")) {
          found++;
        }
      }
    });
  }

[..]
```

Writing a basic module

This guide will explain a very simple example of a basic module. The module is located at scripts/scanner/modules/injection/header-inject.js.

As with response processing modules, every basic module has a metadata object called "module":

```
var module = {
  name: "HTTP Header Injection
checks",
  category: "Injection Modules"
};
```

The entry point of a basic module is a function named initialize() which accepts a single parameter, the "context" object. As with response processing modules, the context object connects the module to Vega, exposing the API. The context object for basic modules is distinct from the object of the same name for response processing modules.

```
function initialize(ctx) {
```

Vega populates a tree-like data structure known as a path state as it crawls a website. Basic modules run on path state nodes, which may either be files, directories, or parameters.

Vega handles identifying and iterating over the parameters while still providing great flexibility to the module developer.

In a very simple example, the following API function accepts only fuzzed parameter values and requires no knowledge of where in the application the module is. This module attempts to inject values into the application. The callback function examines the headers of the responses to try and identify instances of header injection.

The first parameter to this fuzzing example is the callback function, listed in the next code snippet, followed by an array of parameter values to be injected. The last parameter is an optional Boolean indicating whether the injected values are to be appended to a parameter seen by the crawler, in this case it is set to true:

```
  ctx.submitMultipleAlteredRequests(process, ["bogus\nVega-Inject:bogus",
"bogus\rVega-Inject:bogus"], true);
}
```

There are many functions for generating new requests exposed through the context object for basic modules. The reader is invited to view them all at the Subgraph documentation website:

https://support.subgraph.com/trac/wiki/BasicModuleContext

The path state node of the module is accessible through the context object, if the module writer wishes to know information about the path state node on which it is running. The path state structure is explained here:

https://support.subgraph.com/trac/wiki/PathState

When a basic module queues requests for the crawler (for example, to fuzz parameters), it must register a callback function that Vega will run for each of the responses. The callback function is passed three parameters by Vega: the request object, response object, and context object. In this example, the callback function checks the response headers to see if injected values are present, generating an alert if it finds them:

```
function process(req, res, ctx) {
  if (res.hasHeader("Vega-Inject")) {
    ctx.alert("vinfo-header-inject", request, response, {
      message: "Injected Vega-Inject header into response",
      resource: request.requestLine.uri
    });
  }
```

More advanced analysis

The basic module explained in this tutorial is among the simplest examples. It is possible for basic modules to send many requests to perform more complex logical analysis.

One facility for doing this provided by Vega is page fingerprinting, where Vega distills page contents to a simpler representation such that two fingerprints can be efficiently compared for page differences. This is used as the basis for determining positive or negative results in several injection modules, including blind SQL and shell injection. It is also possible to analyze the timing of responses. Readers interested in experimenting with Vega modules are advised to read the API documentation for basic modules on https://support.subgraph.com.

It is useful to note that the ctx.debug() function can be used to print output to the console during module development. It is also possible for modules to store data in a key-value database to share data between modules. The reader should refer to the documentation on the context objects for more information.

Alerts

It is possible to create completely customizable alerts. The template structure is very simple. The example XML file "test.xml" is located in xml/alerts/:

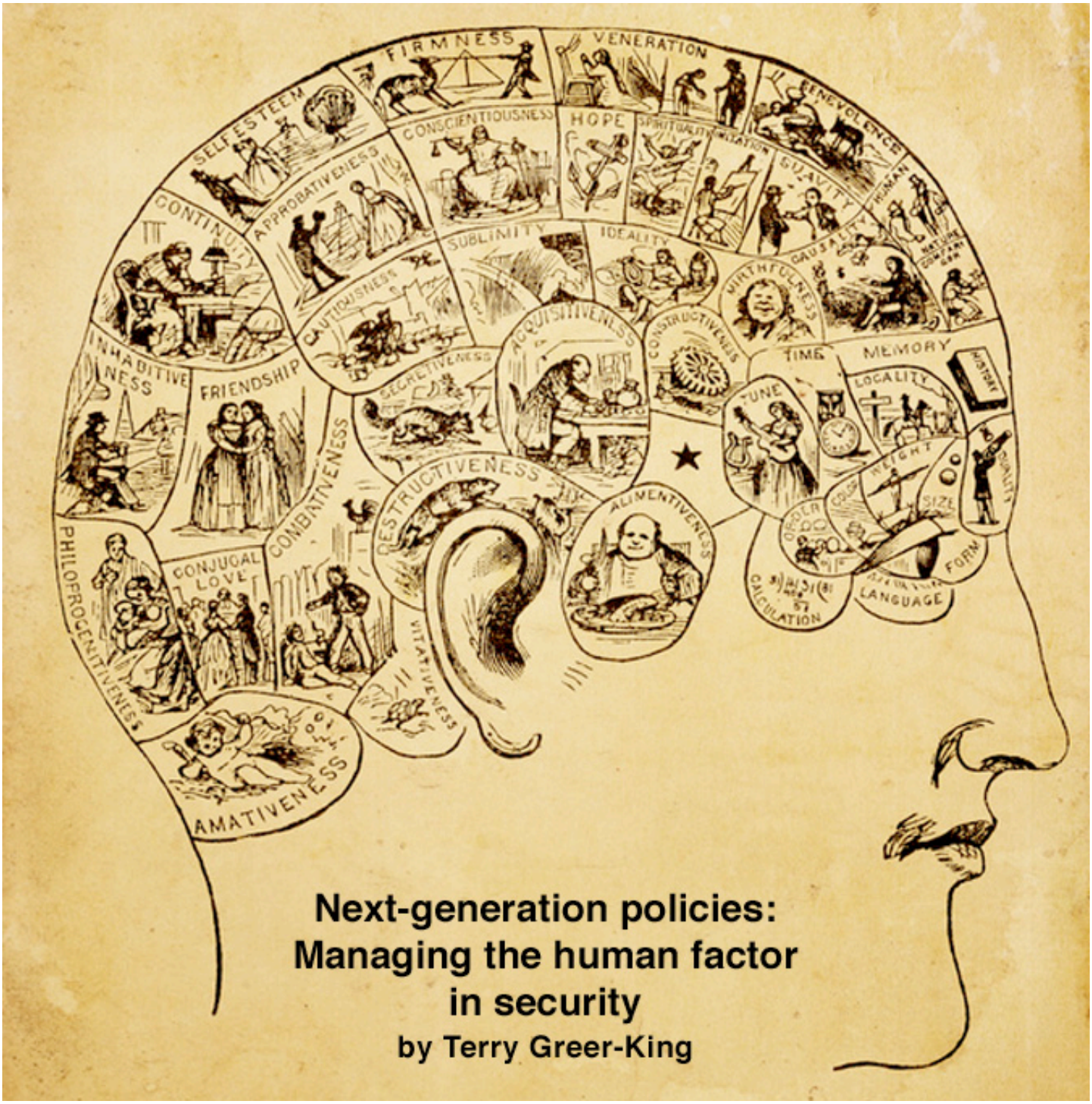
```
<?xml version="1.0" encoding="UTF-8"?>
<alert>
<title>Test vulnerability</title>
        <class>Example</class>
        <severity>High</severity>

        <impact> Could be used to demonstrate partially completed functionality
of web application scanner.</impact>
        <impact> May cause boredom.</impact>

        <remediation>
          There is currently no solution for this vulnerability. Contact your
vendor.
        </remediation>

        <discussion>
                Discuss it here.
        </discussion>

        <external>
                <url address="http://subgraph.com">Subgraph security.</url>
        </external>

        <references>
                <url address="http://minecraft.net">Minecraft is a good
game.</url>
                <url address="http://en.wikipedia.com">Learn stuff here</url>
        </references>
</alert>
```

Conclusion

Vega is a relatively new platform. The primary objective of the project is to build the most extensible platform for web security assessment. The Vega development team hopes to bring entirely new features to support more advanced security checks in the future. We invite feedback via Twitter (@subgraph), e-mail (info@subgraph.com) or on IRC, in #subgraph on freenode.

David Mirza Ahmad is the President of Subgraph (www.subgraph.com). David has over 10 years in the information security business. He started his professional experience as a founding member of Security Focus, which was acquired by Symantec in 2002. David also moderated the Bugtraq mailing list, a historically important forum for discussion of security vulnerabilities, for over four years.

He has spoken at Black Hat, Can Sec West, AusCERT and numerous other security conferences, as well as made contributions to books, magazines and other publications. David also participated in a NIAC working group on behalf of Symantec to develop the first version of the CVSS model and served as editor for the Attack Trends section of IEEE Security & Privacy for over three years. His current obsession is building Subgraph, a Montreal-based open source security startup.



This article shows what next-generation security really means, and why it's critical that organizations understand user and application activity in order to fully protect their networks.

The firewall is now over 20 years old. That's quite an achievement, considering that some security industry observers have been predicting its demise for over half of that time.

Evolving IT infrastructures and increasingly sophisticated security threats have brought repeated warnings about the firewall's impending obsolescence. This started in the late 1990s, when laptop usage and remote access started to spread in the corporate environment, and people began talking about the deperimeterization of networks. A few years later, it was the emergence of SSL VPNs and increased use of smartphones; and today, cloud applications are supposedly the latest threat signaling the firewall's demise.

These predictions usually go hand-in-hand with talk about next generation firewalls, a term that implies we have something new and beyond what came before it.

Certainly, the emerging technology trends mentioned before have forced business networks to handle an ever-increasing number of events and a greater variety of traffic than before.

Border control

While it's true that networks have changed dramatically, from the relative simplicity of a decade ago to far more complex topologies today, and perimeters have become more extended and even fragmented, but those perimeters still exist.

There is still a very clear separation and border between the internal, trusted infrastructure, and external untrusted networks.

Organizations use many different ways to access corporate data, such as client-based and clientless VPNs from laptops and smartphones, or cloud applications – but the borders are still there. Overall network activity is simply more complex, with more events to control, more crossing points, and a greater variety of traffic than ever before.

It's similar to a country controlling its borders. There are many different ways to travel into a country: by air, by rail, by sea or by road – just as there are different ways to access a network. Yet these don't make border security controls obsolete. You simply need to implement different types of controls at airports, ferry terminals and international railway stations, in order to effectively monitor and inspect the different types of traffic.

OVERALL NETWORK ACTIVITY IS SIMPLY MORE COMPLEX, WITH MORE EVENTS TO CONTROL, MORE CROSSING POINTS, AND A GREATER VARIETY OF TRAFFIC THAN EVER BEFORE

What do you mean by "next generation"?

Similarly, gateways have evolved beyond the simple monitoring of certain ports, IP addresses, or the packet activity streaming to and from each address, to be able to scrutinize specific user and application activity.

While this is an evolution, it's not really next-generation. In fact, firewalls have been able to identify applications in-use for the last 17 years by analyzing packet data. Of course, there are far more applications in use in most companies now than ever before, but the principle of application identification in itself is nothing new.

The key issue today is more about adding greater capabilities to look deep within the web traffic passing through the gateway and identify precisely which applications are in use and track exactly which users are running them.

This is the area that's truly new, because companies are no longer just dealing with fixed devices, or static, office-bound users on their networks. Networks have, until now, been defined by the addresses they use. Provided users don't move around too often or change IP address, tracking them is relatively simple. This used to mean that applying security was relatively simple, too, with security policy management defining access based on the internet protocol (IP) addresses of the devices in use. However, this approach to policy management is now dangerously outdated in most organizations, because it is fundamentally dependent on how much physical control you have over the devices that connect to your network.

Moving risks

The growing demand for smartphones and tablet PCs has resulted in employees having multiple devices and, therefore, many IP addresses.

The rise of mobile computing, together with new online applications, makes it difficult for businesses to keep up with policy change requests. If the requests keep coming in based on users and their devices, and organizations are still compiling policies based on static IP addresses, the business is already at risk of exposure.

Even more worrisome is the fact that many of these devices are being brought in from users'

homes without being validated, secured or even looked at by the IT department.

Users are bleeding their personal devices such as tablets, smartphones and personal laptops all over networks, taking work home and bringing home to work.

As organizations adopt more agile computing solutions, they are finding that security policies cannot keep pace with the changes, creating all sorts of headaches. As such, what's needed to help companies manage risk, protect data, audit network activity and give better control over what users are doing isn't a "next generation" product or feature set: it's next generation policies and policy management.

So how should you approach the development of security policies that reflect the way networks are being used today? And how do you ensure those policies are enforced?

User ID checks

Knowing who your users are is critical to managing policy; knowing what IP addresses they are using is less so. As such, defining policy based on user access AND type of device is the only logical choice, as it gives a smarter means for managing access from fast-growing consumerized estates, where the device may not always be known.

Managing devices

Understanding what devices employees are using for network access will also help organizations make informed decisions about their security policies. This allows them to track what devices have accessed which data, so if they need to determine where the networks may have been breached from, there is already a defined limit on the number of people and devices with access. Consider just how more effective a security policy could be with the addition of this parameter.

Application control

The ability to identify application activity on a firewall or gateway is nothing new. However, the ability to identify applications that are NOT defined by standards - such as web applications, social media portals and more - is a powerful addition to creating a next generation policy.

If you add the ability to detect and manage user access to those applications, businesses can further strengthen application control. By allowing users to interact with the security system, both to remind them of corporate policy on acceptable use of applications and to take feedback in real-time on why the user needs access and the intended purpose of their usage, organizations can add a further layer of security reinforcement and protection.

Data – the core element

The three points covered so far help to ensure organizations can identify which users are accessing the network, from which device or application. However, the core element of security policy is the ability to analyze the data that is being accessed, sent and manipulated to ensure users are not sharing – or leaking – sensitive information.

This requires assessing not only what applications employees can use, but what data these applications are allowed to use, and, in turn, taking steps to protect sensitive data from inappropriate or non-compliant usage.

In conclusion, the increasing adoption of consumerization, virtualization and cloud computing means that network infrastructure is no longer static: it's agile, dynamic and fragmented, with data flowing in unexpected and unpredictable ways.

Next-generation security has to include the "human factor" – the people using networks, the devices they use, the applications they are allowed to run, and the data those applications can access and modify – to reflect this dynamic network usage. Only then can you create cohesive, next-generation security policies that truly protect what matters to your business.

Terry Greer-King is the UK managing director at Check Point (www.checkpoint.com).

CYBER DEFENCE SUMMIT مؤتمر الأمن السيبراني

ENDORSED BY

APRIL 2ND - 3RD 2012
GRAND HYATT HOTEL, MUSCAT, OMAN
WWW.CYBERDEFENCESUMMIT.COM

DEFENDING YOUR VIRTUAL BORDERS

MIDDLE EAST IS GATHERING TO DEFEND IT'S CRITICAL INFRASTRUCTURE

TELECOM & IT SERIES

PLATINUM SPONSOR

GOLD SPONSORS

SILVER SPONSOR

BRONZE SPONSORS

MEDIA PARTNERS

For more information on being a part of this summit, contact: **Ali Khalid Rana,** Marketing Manager
Email: alir@cyberdefencesummit.com, **Tel:** +971 4 455 7962