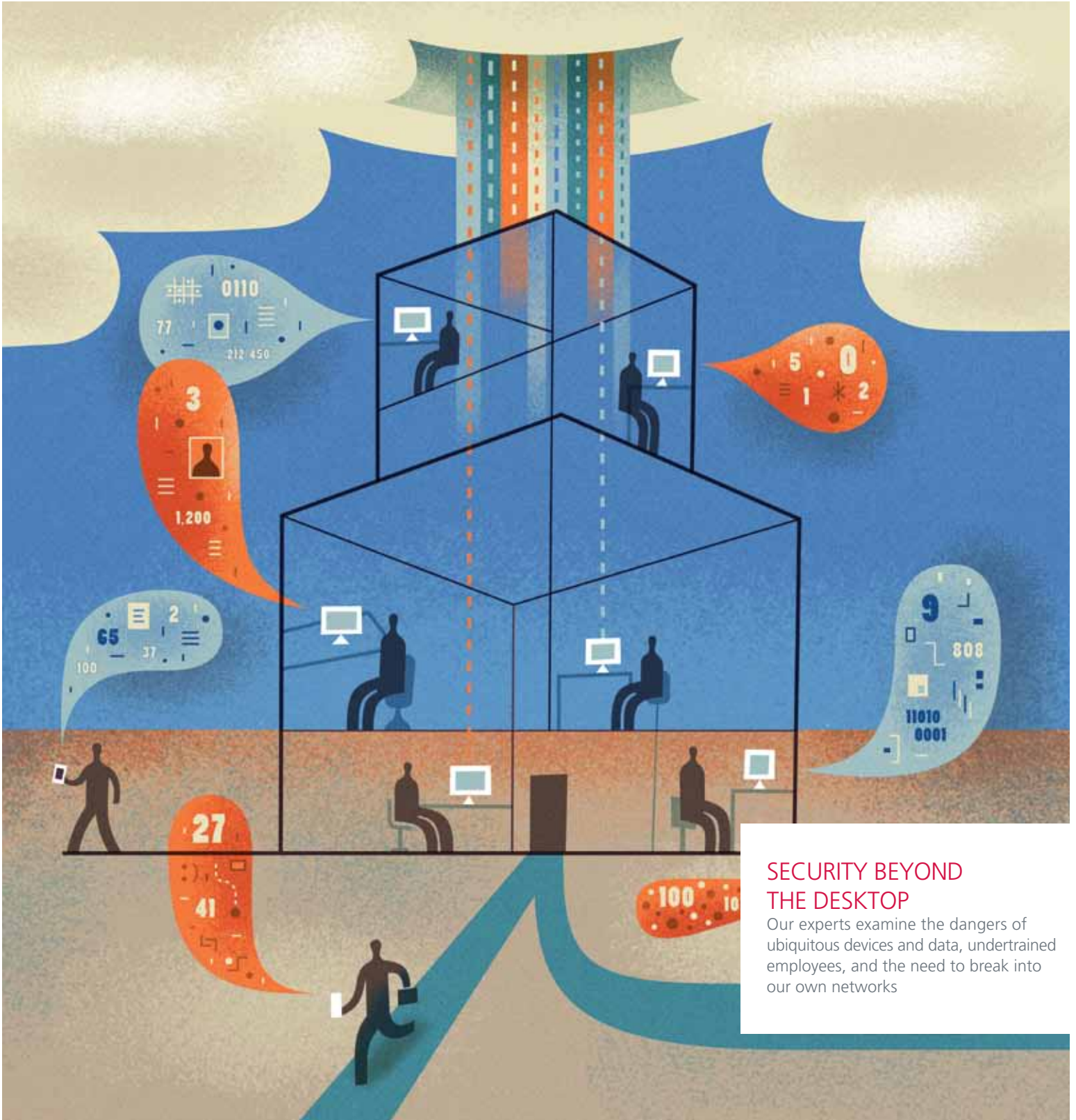


Security Journal

Issue 7, 2011



SECURITY BEYOND THE DESKTOP

Our experts examine the dangers of ubiquitous devices and data, undertrained employees, and the need to break into our own networks

Contents



- 3 **Security Industry Needs a New Vision to Conquer Current and Future Challenges** Device and connection ubiquity, user behavior, poor security preparation and design, and the cloud have pushed information security issues to the forefront of world events. **By Vincent Weafer**
- 5 **Security Is Often an Illusion in Today's Access-From-Everywhere Climate** We need to improve software development to avoid vulnerabilities, better train employees, and improve our procedures. **By Chris Roberts**
- 8 **Improve Intrusion Prevention by Investing in Employees** We rely too much on technology to keep our data secure. We need to educate, empower, and encourage our human "intrusion prevention systems" to greatly improve our security. **By Jayson Street**
- 11 **Mobile Privacy Risks Fuel Future Threats** Mobile phones contain personally identifiable information and corporate data. We need to do a better job of protecting both. **By Jimmy Shah**
- 16 **Social Engineering Eases Real-World Penetration Testing** Many attackers bypass the Maginot Lines of security systems by using social engineering tactics to talk their way past employees. **By David Kennedy**
- 20 **The Cloud Is Changing Everything** Moving to the cloud means the network perimeter has disappeared. The bad guys have noticed and are now targeting the cloud to worm their way into organizations. **By Scott Chasin**
- 24 **SQL-Injection Attacks Too Easily Threaten Data** Many of today's big security breaches come via the relatively simple hack of SQL injection. We need to improve our defenses, which will drive most attackers to look elsewhere for easier targets. **By Vadim Pogulievsky**
- 28 **Can Studying the Past Help Secure the Future?** A look at the history and motivations of hacktivists can help us improve security. We need to reexamine the basics of security and anticipate attacks using penetration testing and other methods. **By David Marcus**

McAfee Security Journal

Issue 7, 2011

Senior Vice President, McAfee® Labs™

Vincent Weafer

Director, Security Research and Communications

David Marcus

Editor

Dan Sommer

Authors

Scott Chasin
David Kennedy
David Marcus
Vadim Pogulievsky
Chris Roberts
Jimmy Shah
Jayson Street
Vincent Weafer

Marketing

Marshall Cannon
Judith Kemp
Beth Martinez

Copy Editor

Mary Karlton

Design and Layout

PAIR Design, LLC

Artist

Doug Ross

Printing

Xerox Global Services

Translations

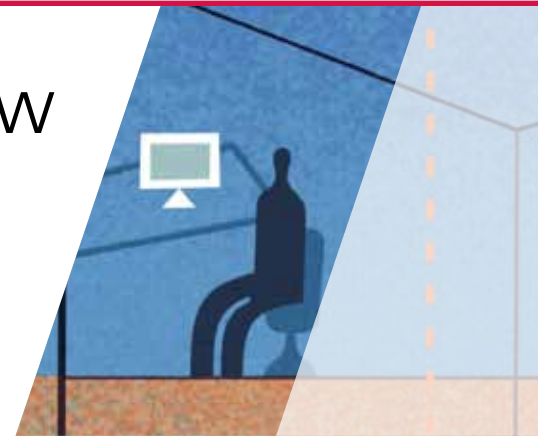
McAfee Localisation
Nancy Matis SPRL

Public Relations

H3O Communications

Security Industry Needs a New Vision to Conquer Current and Future Challenges

By Vincent Weafer



Welcome to the 2011 edition of the *McAfee Security Journal*, the publication that reflects our security vision. One of the things that drew me to McAfee Labs last year was the chance to develop a new vision.

Businesses, enterprises, users, and governments are more connected than ever through technology, which is a great leveler and distributor of power. But great power brings great responsibility. There are more interconnections and interconnected technologies than ever before, and these have led to a quantum leap in security, data, and privacy concerns. Security is no longer solely a host or network issue. Nor is it solely a cloud issue. The issues of information security from this point forward are systemic issues. Insecurity is now engineered into our systems by the nature of their many ubiquitous connections — whether they be local, mobile, embedded cloud, or any other kind — as well as by the behaviors of our varied types of users. Security has truly moved beyond the “desktop”; in this issue of the *McAfee Security Journal*, we tackle these topics head on with an exciting lineup of writers and researchers.

Chris Roberts, founder and Chief Geek of One World Labs, who you may remember from the McAfee “StopHCommerce” videos, leads off. He explains, using his provocative yet humorous style, that today’s access-from-anywhere technology is way too complicated while security is inadequate and humans form the weakest link. Conveniences such as free wireless, weak passwords, and Bluetooth

often combine to form systemic issues that keep us at risk. Roberts reminds us that we need better security development, better-trained employees and users, and better overall procedures. Next, we have Jayson Street, author of *Dissecting the Hack*. As he explains, we rely too much on technology to keep our data secure. Street calls for better training, empowerment, and encouragement of our “human intrusion prevention systems” to greatly improve systemic insecurity. He strongly lays out an argument that well-trained employees will be far less likely to fall victim to social engineering attacks.

Attackers Turn to Mobile Devices

Our lineup continues with Jimmy Shah, one of the most senior mobile threat researchers at McAfee Labs. He describes mobile privacy and its future threats. Mobile phones contain both personally identifiable information and corporate data, and we need to do a better job of protecting both. Mobile devices go beyond cell phones; tablets and embedded devices could also be vulnerable. How about a hacked automobile? David Kennedy, creator of *The Social Engineering Toolkit* and experienced penetration tester, reminds us that it is still too easy to break into corporate networks. He shows how effective social engineering can bypass today’s complex security systems, proving them to be little more than digital Maginot Lines. Why butt heads with potentially strong hardware and software protections when hacking the human condition is far more effective? Companies that test themselves for weaknesses before cybercriminals do will become much more secure.

Scott Chasin, CTO of cloud security at McAfee, discusses how life in the clouds is changing everything because the network perimeter has effectively disappeared. The bad guys have noticed this transition and are now targeting the cloud to worm their way into organizations. As Chasin argues, the cloud has advantages: primarily reducing IT costs via its service model, but the challenge is to ensure these services are secure as well as accessible. Next is Vadim Pogulievsky, security research manager at McAfee Labs, who shares his thoughts on the relative ease and prevalence of SQL-injection attacks. Many of today's big security breaches come via the relatively simple attacks of SQL injection, a threat that, in spite of its simplicity, is still not effectively blocked. Inadequate web design plays a big role in enabling this threat vector. As more and more services migrate to the cloud, attackers may find that this is one of their favorite pathways to success.

Learning From Hacktivists

We close with David Marcus, director of security research at McAfee Labs. Using examples of recent and past hacktivist activities, he urges us to reexamine the basics of security and change our perspective. We must expect to be attacked. By looking at history and applying a little order and method, our industry may learn some useful lessons from hacktivism.

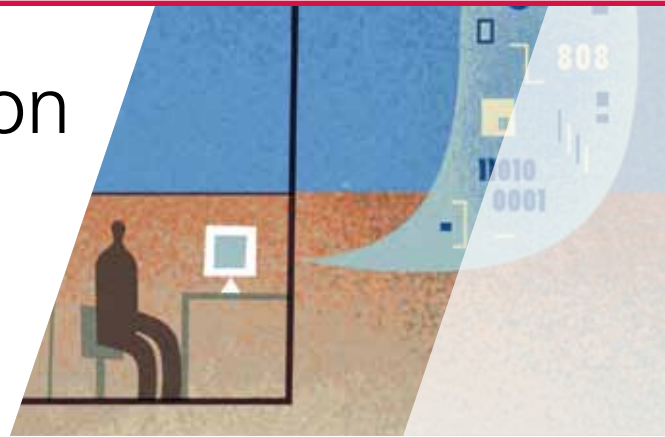
The threat landscape has evolved steadily over the years, but never more so than during the last two. Device and connection ubiquity, user behavior, poor security preparation and design, the cloud, and even the apparent ease of today's hacktivist attacks have pushed information security issues out of the cellar of IT to the forefront of world events. We need only to read the news to see that security has moved well beyond the desktop. We must push our industry to meet these new challenges if we are to protect what people and enterprises value.



Vincent Weafer is senior vice president of McAfee Labs. He manages more than 350 researchers across 30 countries as well as millions of sensors around the globe, all dedicated to protecting our customers from the latest cyberthreats. His team works to advance research and intelligence-gathering capabilities to provide the latest protection solutions in malware, host and network intrusion, email, vulnerabilities, regulatory compliance, and web security. Weafer's experience comes from more than 25 years in information technology, ranging from software development, systems engineering, development management, and security research positions. Before moving to McAfee Labs, he was the leader of Symantec's Security Response team, a position he held for 11 years.

Security Is Often an Illusion in Today's Access-From-Everywhere Climate

By Chris Roberts



The technology we use today is central to so many areas of our lives. We understand some of this technology, but there is much we do not understand.

We have successfully managed to build complex systems that no single person understands, and we have evolved the technology to a place where few could have imagined; yet we suffer from the ever-increasing feeling that we're losing more knowledge than we're gaining. (During several on-site client visits, we have heard: "Nobody knows what that old system in the corner does, but we know it's needed.")

Our computing use has gone so far beyond the data center that I think I can prove my microwave has more processing power than the ICL and VAX systems I used to work on, and I certainly couldn't have stuck the VAX in my bag and fired it up at the coffee shop! Today, we interact with data through the extended network that we have built up during the past few years—and now we have to deal with support and security issues as everyone clamors to connect to the enterprise architectures in our care. This convenience, unfortunately, takes the data we've been able to manage and control over the years into the public space, which, as we know, is not exactly a friendly environment.

I'm not sure if I should be scared about moving our mail, systems, accesses, and data sets to mobile devices. After all, we've been doing that on laptops for years. In my case, I've had the same laptop for four years, though I've gone through eight to nine phones in the same time—mostly due to accidents. But even so, that's a heap of data "out there," and that does not include the backup laptop, the assessment devices, Apple iPad, and other systems that carry critical information. My gear is well-encrypted or has other protections in place, but I'm the exception to the rule. For the most part, organizations allow personal devices, unencrypted devices, B2B clients, and all manner of other devices to connect to their critical information—often with minimal controls (or validation of controls) in place. Furthermore, a lot of places still think this is "okay."

The Weak Link

I'm a fan of the *Terminator* movies. You have to give credit to the Skynet concept: we, the humans, are still the weakest link in the information security chain, we are susceptible to influence (easily fooled), open to negotiation (bribed), incapable of remembering complex processes (passwords on sticky notes), and distracted ("squirrel!"). We have a problem maintaining contact with our data sources ("Where did I put that phone?"), and, for the most part, we don't fully grasp our surroundings.

It's no wonder that the Skynet system decided to move on without us. If we were computer systems, we'd have been shelved years ago and probably upgraded with a new shiny Apple MacBook. (They don't get viruses, do they?) The problem is that during my time in IT, we have gone through two distinct cycles of data migration (centralize/distribute, centralize/regionalize) and are once again in the middle of heading to the distributed model of keeping our data all over the darn place in the cloud. (The cloud resembles the closets we had as teenagers. We shove everything in there, assume all's well, and go out with the important stuff stuck in the pockets of our trousers—though today that stuff resides on a dozen unprotected devices.)

What, Me Worry?

Let's take Skynet a little further. Say that we are given a chance to "justify our existence" in the electronic world. There are potentially several criteria for being a good custodian. Two are key. Are we responsible? Heck no, we can't even manage to look after the data when it's locked in a data center. How are we going to survive when the iPhone/Droids/Crackberries with all of our emails, passwords, and documents keep getting lost/stolen/misplaced/flushed



down the bog? Are we conscientious? Good grief, no! We pride ourselves on passing audits and fooling auditors. I sometimes wonder whether we give a fig about the moral obligations of looking after customers' personal data—especially if it impacts the profitability of the company. We spend more time arguing whether a breach should be disclosed than actually fixing the problem, and, as individuals "outside" of the problem, we, as customers, have become so oblivious to the actual issues surrounding security that the banks have removed all fiscal responsibility from us. The identities that are lost/stolen/misplaced every hour barely make the news, and, if they do, we've seen so many of them that our reactions are muted.

Those who do care are typically owned, controlled, or managed by those who don't. Those in power are influenced by the same groups or lobbyists whose job is to ensure we don't care. If anything ever makes it as far as regulation, it becomes so watered down that it's almost worse than useless. As humans we pretty much suck as responsible guardians for anything data related. You can almost feel sorry for Skynet as it becomes "aware" and realizes we're such a threat. If you want to argue, go ahead; but watch a couple of documentaries first. "Enron: The Smartest Guys in the Room" would be a perfect start.¹

Back to Basics

Let's spend a moment on the basics that we have in place today and see how those core elements will translate into the portable, ever-expanding world beyond our desktops.

We're greeted in the morning with a password prompt to log into the computer. Eight characters and we're done, unless you are part of the 5 percent who encrypt their machines (in that case, congratulations for being security minded and for remembering two passwords). At this point, the machine's anti-malware program wakes up and begins its sentry duty, theoretically monitoring all

ingress and egress points. We traditionally begin our day secure behind firewalls, intrusion detection and prevention systems, spam filters, data loss prevention, and all manner of other devices designed to protect us from the malicious world. It seems pretty safe. However, in our current untethered world, I grab my portable device, enter a four-digit passcode (that's probably the same as the PIN on my debit card), and start to surf the web and download mail, apps, and all sorts of other things without a care in the world. I do this with a Bluetooth device stuck in my ear listening to the conference call over the free wireless VoIP network, thus bypassing every control known to humankind. (For those of you complaining about passwords, try typing eight alphanumeric and special characters while driving. If you haven't crashed yet, you'll be yelling at the IT team to amend the information security policy.)

There's something rather scary about that picture I've painted, but it's repeated in offices, business parks, and homes all over the world. Yes, I've taken some of the images to the extreme, but we security professionals come across these scenarios daily. And the situation is not improving. Users demand flexibility, connectivity, and ease of use from our mobile devices, and we want to avoid the constraints that are seen to tie down desktops and laptops. We are simply trying to do too much, with too little control, and this ambition is going to come back and bite us.

We are simply trying to do too much, with too little control, and this ambition is going to come back and bite us.

Fooling Ourselves

I believe we are frequently in situations where, for the most part, security is an illusion. We seem to think it is a good thing to pass compliance initiatives by fooling the auditors rather than using them as tools for change. We have more oversight than ever. We have more rules, regulations, policies, procedures, and controls in place than we know what to do with. Yet we are still being breached, losing data, and managing to make some pretty glaring mistakes. We demand more and more of our staff, and we expect them to be perfect. We throw more hardware at the problem, and we have monitors monitoring the monitored systems producing all manner of reports that gather dust and fill up the shelves. Meanwhile, we continue along this path despite evidence that we're heading in the wrong direction. My view is not that of a jaded and cynical security professional; this is the view of a seasoned leader of a team that typically breaks into and "owns" the client by lunchtime on the first day.

We are at a point where succeeding with a penetration test against an organization involves little more than gaining access to a single laptop/desktop/portable device within the infrastructure, planting a Trojan (via hardware or software), and getting out undetected. The problem is that with the proliferation of computing beyond the desktop, many of those targeted systems containing critical data (or access to it) don't live behind the traditional locked doors that are guarded and managed. We can sometimes take out the core infrastructure of a targeted organization while sitting at the coffee shop. (If you think the coffee shop is a dangerous place, consider sitting on an airplane for three hours. I have a captive audience who will happily associate their devices with my "free Wi-Fi"—just you, me, and a password cracker for several hours.)

Let's take a moment to think about the forensic world, which we enter when, eventually, something goes wrong. In the old days, a man in a suit arrived at your office, put on his latex gloves, and proceeded to do unfathomable things to your computers. These days the poor guy must chase around the building looking for laptops, PDAs, cell phones, smart devices, tablets, portable drives, and USB sticks. He invariably finds that half the equipment in question is at someone's home, whereupon he's dispatched to collect all the devices and probably image the home systems because the company lets its employees use personal computers for corporate activities. He also must collect logs from every conceivable network and security device, and we won't talk about needing to image servers that reside "in the cloud" in a foreign country.

A Challenge

I've painted a pretty gloomy picture so far, and with good reason. We face a multibillion-dollar underground enterprise that operates solely to steal, launder, and resell the data we are trying to protect within the current (somewhat centralized) corporate and government environments. At the same time we are embarking on a journey to disseminate that information, and the secured access to it, to the very edges of our control. We needn't despair, however. We need to go back to the basics: the code (security as part of the software development life cycle, not an afterthought), the people (awareness training), and the policies, procedures, and controls we should have in place. We need to revisit what we missed the first time and fully implement the stuff we planned to do. If you don't re-secure your systems for the sake of your company, the auditors, or your customers, then do it for my sake. On the first day of penetration testing I shouldn't already "own" the Active Directory Forest and sit with the CFO's credit cards in hand. Do me a favor—make it a challenge.



Chris Roberts is the founder, CISO, and chief geek of One World Labs, an assessment, remediation, and research facility in the Front Range area of Colorado. He has played a variety of roles in organizations and as a consultant to the IT security, engineering, and architecture/design operations of a number of Fortune 500 companies across the finance, retail, energy, and services sectors. Roberts has a wealth of experience conducting vulnerability assessments, penetration testing, compromise investigations, and digital forensics examinations of all types of information systems. He can typically be found as the tall, bearded one wearing PJs around the client site, manically giggling while holding a laptop in one hand and coffee in the other.

ENDNOTES

- 1 http://en.wikipedia.org/wiki/Enron:_The_Smartest_Guys_in_the_Room

Improve Intrusion Prevention by Investing in Employees

By Jayson Street



Here's a common situation: after weeks of vendor comparisons, proof-of-concept testing, purchase approvals, and installation, you finally have that shiny new intrusion prevention system (IPS) installed in the data center.

You have the best technology and the best implementation design possible to protect your company from the legions of threats outside your network.

Yet six months later, you find yourself sitting in front of the CEO explaining how part of your company's customer database walked out the front door on a USB stick after being downloaded by someone impersonating an employee.

If you think technology can solve your security problems, then you don't understand the problems, and you don't understand the technology.

—Bruce Schneier, Chief Security Technology Officer, BT and author

That quote from security maven Schneier is so often used in the security business, that it is now cliché. But that's because it is so true. Security professionals install systems and layers and monitoring—and bad things still happen. Often, when bad things happen, we lament the “dumb users.” Security professionals who think that way need a new point of view—it is not enough to tune intrusion detection rules, update IPS and anti-virus signatures, and patch systems. Your offices are full of human intrusion prevention “systems” sitting at desks waiting for you to educate, empower, and encourage them!

Educate

I have worked in banks before and have seen how they train their staff to deal with robberies. If someone walks into a branch wearing a ski mask and passes the teller a note demanding cash, every employee knows what to do. In fact, they practice what to do in such scenarios.

But if I walk into the same branch and claim to work for the utility company, I can probably get behind the teller line on the pretext that I am checking for “recurring low-voltage problems on the secondary phase inducer.” (My excuse doesn't have to make sense.) Most banks do not teach their employees how to deal with such social engineering attacks.

Your task is to teach your people what to do. They are smart people! They wouldn't have their jobs if they were incapable of learning skills, and this is just another skill that they can learn. Are you concerned that it will take time? You take the time to update the anti-virus signatures on all your servers. You take the time to patch everything in your operating systems. You even tune the signatures on your IPS. But do you educate your most important defense systems—your employees? Overlooking employees is no different than leaving that shiny new IPS spinning in the data center with the same signature set it was shipped with a year ago.

One of the best ways to prepare employees for events such as social engineering attacks is to train them to ask themselves, “Does this happen in the course of my daily routine?” In my bank example, the utility company doesn’t send people straight into bank branches. They will call headquarters and make arrangements with the facilities staff. Last month, a number of emails in circulation claimed to come from the FDIC. But the FDIC doesn’t communicate with the general public that way. In both cases, the ruse used a scenario that was out of the ordinary. People often know when something is amiss. They just need the confidence to act on their suspicions.

The most important lesson in education is to teach your people that it is okay to be wrong. Give them the tools to report suspicious activity. They need a hotline phone number and an email account—which you had better monitor and respond to, or people will stop using them. When you get reports, thank the person who contacted you. Even if you get a false alarm, verbalize your appreciation. Reinforce the fact that you need data to do your job. Security works best when people tell us something doesn’t look right. And every “thank you” you hand out is another tuning of the human IPS sensors who are sitting in chairs throughout your organization.

Empower

The job description for every employee in your organization should include a statement about information security. No matter what role someone has in your organization, he or she touches the process of protecting company and customer information in some way. From new-hire orientation to annual training to content on the company intranet, remind everyone how important this job is in every context. And remind them that it is a part of their jobs. The technology team can’t do the job alone. The compliance team won’t get it done either. Everyone has to work to protect the company.

I have seen this reluctance to question suspicious behavior during penetration tests. I faced an employee with a story so full of holes, there was no way I expected to get past my first challenge. I remember hearing the voice in my head almost cheering for the employee. “Don’t do it. Please. You seem like too nice a person for me to have to include you in the report. Don’t open the door for me. Doh!” And I was in.

After I walked out of the data center and explained who I was, I asked the employee, “Why did you let me in?”

“I figured if you got in this far, you were already authorized.”

“But I could see in your face that you didn’t believe me! What happened?”

“I don’t know. Sorry.”

That person should have been told early and often, “You are the first line of defense! You have the power to make a decision. In fact, even if you make the wrong one, as long as you err on the side of protecting customer and company, we will stand by you.” Empower your people! (Yes, I am repeating myself because we all need to hear this.)

Encourage

Enforcing a policy is not a negative. Don’t turn *enforce* into *browbeat* or *gotcha*. Enforcement is walking around with a pocket full of gift cards and dropping one on an employee’s desk when you see her practicing good security. Enforcement is writing up a praise-filled story in the company newsletter about someone who took the time to do the right thing.

There are ways to run inspections that are both enforcing and educating without turning into “gotcha” audits. For example, physical security habits are a fundamental requirement of good information security. If employees are in the habit of leaving workstations unlocked and passwords written on notes by their monitors, walk around. Talk to people. Perhaps set up a process of recurring inspections. But do it publicly. Let people know what you are looking for. Praise people who are doing a good job. The word will spread, and employees will start adopting better habits.

Enforcement is walking around with a pocket full of gift cards and dropping one on an employee’s desk when you see her practicing good security. Enforcement is writing up a praise-filled story in the company newsletter about someone who took the time to do the right thing.

When you get an automated alert of a response action by your anti-virus system (you had better already have that configured), show up quickly. Make sure everyone knows the security team is notified when things like that occur. Talk to people about how malware gets on computers. You are not just talking to the person with the malware alert, you are talking to every person he is going to tell about his experience with the security team.

We will continue to read about social engineering and other intrusions: another defense contractor here was compromised, and a security vendor there was breached. Look deeper into the story, and you will see it was not the technology that failed. It was the people. Did they fail because they were reckless? Were the workers unintelligent? Or did someone fail those workers (and their customers) by not adequately preparing them for these social-based attack scenarios? The way you avoid that situation is by educating, empowering, and enforcing the security habits your team needs to survive in this threat-filled world. Remember, your “team” is your entire company. Congratulations, I bet you didn’t realize you had such a large staff. Now start acting like you do.



Jayson E. Street is coauthor of *Dissecting the Hack: The Forb1dd3n Network*, from Syngress, and the creator of the community site <http://dissectingthehack.com>. He has spoken at DEF CON, BruCON, uCON, and several other -CONs and colleges on a variety of information security subjects. Street was chosen as one of *Time Magazine's* persons of the year for 2006. Twitter: @jaysonstreet

Mobile Privacy Risks Fuel Future Threats

By Jimmy Shah



I deal with mobile malware every day, and my friends will occasionally ask me if their smartphones are safe.

They assume that I'll tell them whether the latest game or specialty app they downloaded is going to destroy or "brick" (render unusable) their phones. I inevitably tell them it's not that simple anymore.

These days your privacy is under siege from attackers. They have a variety of ways to make money by stealing personally identifiable information, photos, and other private data. A smartphone contains a lot of data—your name, identification numbers, addresses (physical and electronic), and pretty much any other information that can identify you. This type of information is useful to both advertisers and identity thieves. The former will probably use it only to sell you their legitimate products, but the latter want to profit at the expense of your credit.

A broken or unusable smartphone does not make any money for attackers, but swiping personal information gives them the gift that keeps on giving. An attacker can use your information either to apply for credit cards or to access other accounts with stored money (for example, Skype or IM). If that is too much work, they can bundle your personal data with that from multiple other victims and resell it to larger-scale criminals or organized crime.

Privacy-Invasive Apps

There is a scale of mobile threats to privacy, ranging from the simplest—which require only the installation of a standard app—to the complex—which exploit vulnerabilities in the operating system to gain complete control over the smartphone. OSX/iPHSponey.A is one example of the simple kind.

OSX/iPHSponey.A was a proof-of-concept Apple iPhone spyware application written by developer Nicolas Seriot.¹ It is designed to use only Apple-authorized APIs. His goal was to produce an application that uses only allowed functions to gather as much data as possible from the phone.



Figure 1. Spyware app OSX/iPHSponey.A collects personally identifiable information from multiple iPhone apps.



Figure 2. Personal data is emailed in a report to the attacker.

His app was able to gather data from a number of other interesting apps, such as Safari, YouTube, Mail, Address Book, and more. The data was then emailed to the attacker. OSX/iPHSponey.A was also able to access the keyboard cache, which holds pretty much every word you've typed on screen. You've entered passwords, love letters, and other sensitive information into the browser and other apps, and this spyware can grab a copy of all of it.

The actual threat from this spyware was limited because it was distributed only as source code and the email reporting address was hard coded. This would make tracking an attacker and shutting down an account relatively easy.

Another simple attack was J2ME/VKonPass.A, a Java app that pretended to be a mobile client for Vkontakte, a popular Russian-language social networking site. In reality, this was a phishing app. When a user enters his username and password to log in from a smartphone, the app displays a communication error while in

the background, it emails the login credentials to the attacker. Because the email address of the malware author was encoded in the app, it would again be easy for authorities to shut down an attacker.

Social Network Session Hijacking

Late last year, security researcher Eric Butler released an extension for the Firefox web browser called Firesheep.² This extension allowed an attacker to hijack the web browsing sessions of unsuspecting victims.

The tool didn't originate the attack—it just eased the process by creating a point-and-click attack. The attacker can sit down at a coffee shop with a laptop and wait for people to log in to Facebook, Twitter, or other services. Firesheep then allows the attacker to access the accounts with the credentials of the victims.

Recently, researcher Bartosz "ponury" Ponurkiewicz produced a similar tool, FaceNiff, for Android smartphones.³ This app lies near the middle to high level of complexity on the mobile privacy threat scale.

Just like Firesheep, FaceNiff monitors your local Wi-Fi connection and waits for logins to popular social networking and shopping sites such as Facebook or Amazon. The emergence of an Android app that helps steal accounts lowers the bar for the casual attacker. To log in to an account, the attacker needs only to press on a captured account.



Figure 3. The phishing app J2ME/VkonPass.A steals login information and emails it to the attacker.



Figure 4. With FaceNiff, hijacked sessions are listed by username. One click loads the site in the attacker's browser.



Figure 5. After choosing an account with FaceNiff, an attacker can load a profile and do everything but change the password.

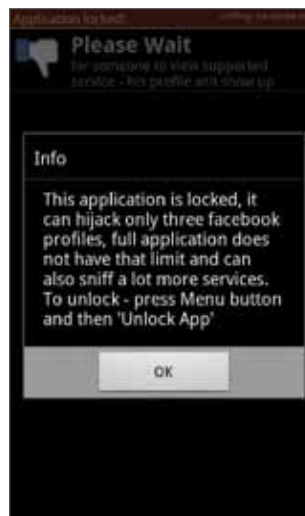


Figure 6. Registering FaceNiff gives an attacker access to additional social networks or websites.



FaceNiff by default allows an attacker to hijack only three Facebook sessions. Ponurkiewicz sells access to more social networks for a small registration fee.

FaceNiff for Android is more complex than OSX/iPHSponey.A, but its threat is still small. FaceNiff requires a “rooted” Android phone to access the network. (See the next section for more on rooting.) There are also a number of steps users can take to prevent session hijacking:

- Use the secure login versions (<https://>) of websites
- Use Firefox and install the Electronic Frontier Foundation’s HTTPS Everywhere extension⁴
- Use a virtual private network

Jailbreaking, Rooting, and Reduced Security

“Jailbreaking” (for iPhone) or rooting (Android) your smartphone opens it up to more functionality and customization, but it can also reduce system security. Attackers have, in the past, created worms to log into jailbroken iPhones and modify the wallpaper

or enable phishing schemes. Malware authors have written advanced Trojan horse programs that use root access on Android phones to prevent malware from being removed.

Malware authors have not yet fully exploited the reduced security on a jailbroken or rooted phone. Some advanced work is being done by security researchers. Eric Monti demonstrated an advanced attack on a jailbroken iPhone at Toorcon last year. He used a modified version of the Jailbreakme.com exploit to silently install a rootkit on the phone. Once it was installed, he logged in and spied on a debit card transaction being handled by a popular credit card processing app. Because he had root access, he could easily have stolen all contact information, email, SMS messages, and passwords.

An attacker gaining root access on your smartphone and installing a rootkit is at the top of the mobile privacy threat scale. Fortunately, malware authors and attackers are not currently developing their own root exploits.

Risks of Software Piracy

Software piracy hurts more than just software developers. Pirated software has been a large vector for new mobile malware and spyware.

Ponurkiewicz, the author of FaceNiff, built a tool that can be used for hacking accounts, and he is trying to make a little money by selling extra functionality. Like a developer of legitimate software, he has come face to face with software piracy. Within three days of releasing the latest update to FaceNiff, some enterprising software cracker modified the app and released a version that does not require buying an activation code from Ponurkiewicz.

The real danger of software piracy is to users. We have seen a large increase in malware authors using legitimate apps to hide their malicious code. Instead of writing custom Trojans such as J2ME/Vkonpass.A, malware authors are repurposing games and utility apps. The benefit is that unsuspecting victims will download what looks like a pirated version of a trusted app and end up putting their personal information at risk. In some cases the pirated version might be more popular than the original app. A number of notable Android privacy threats have had great success with pirated apps. These attacks span the scale of mobile privacy threats—some simple and some very complex.

Android/SteamyScr.A is an interesting example of malware that consists of malicious code injected into a novelty app. The real app lets you draw on the screen as if it were a steamy shower door. It is a simple, highly popular app. The malware, however, is complex. Initially, it sends attackers a number of pieces of information (unique identifiers for mobile phones, SIM card ID)

that would allow them to uniquely identify your phone. This is likely so that attackers could maintain a list of unique infections. After that step, it performs other actions such as signing up the victim to a number of premium-rate SMS subscription services. SteamyScr.A forwards all of the victim's contact information to the attacker, and it is also able to install additional software on the smartphone without the user's permission.

Android/Jmsonez.A is malware injected into a real calendar app.⁵ A side effect of having the malicious code within the app is that the calendar no longer functions properly. Every time the calendar launches, it starts at January, regardless of the actual month. And that's not the worst thing it does. As with SteamyScr.A, Jmsonez.A also signs up the victim to an expensive SMS service. To avoid detection, it deletes any confirmation messages that the subscription service sends to the user.

Subscription SMS services are a threat to your privacy beyond the cost of the messages. The services will usually get billing information for the victims that may include addresses, which they can use to target the victims again or resell to other attackers.

The authors of Android/DroidKungFu.A go a step further than the previous malware.⁶ With single pirated apps, you can just avoid them and not worry. DroidKungFu.A takes hints from other complex Android malware families, such as Geinimi and Android/DrDread—its code is included in a number of apps, including a handful of games. Like DrDread, DroidKungFu.A also uses a pair of root exploits to reduce system security and maintain itself on the device.



Figure 7. Android/SteamyScr.A pretends to be an app that lets you draw on a steamy shower door. While you're doodling, it can steal lots of information.



Figure 8. Android/Jmsonez.A tries, though poorly, to pose as a legitimate calendar app.



Figure 9. One of the many legitimate apps infected by Android/DroidKungFu.A.



Figure 10. The Carwings service can leak your current location and your destination to information providers.

Because DroidKungFu.A can gain root access to your smartphone, it can easily grab any personal data (contacts, SMS, email, and more) on your system. It can also install adware, spyware, or updated versions of itself, further reducing your privacy. The attacker can command an infected smartphone to visit certain websites, which can lead to ad click fraud, phishing, or further infections.

Information Leaks

Your smartphone isn't necessarily the source of invasions of your privacy. Occasionally, other devices you own can be the culprits. The Nissan Leaf electric car includes a GSM cellular data connection. Essentially, this car has its own smartphone.

The data connection sends information on fuel economy to the manufacturer and receives news updates from various RSS feeds. (See Figure 10.)

For fuel efficiency reasons, it's useful for Nissan to have your location, speed, direction, and the coordinates for your destination. It's not so useful for a news website to have your car's current longitude and latitude in their web server logs.

The location functionality is intended for the Carwings service. Information providers can use your position to tailor what they send you based on your location (for example, the nearest gas station or rest stop).

ENDNOTES

- 1 http://vil.nai.com/vil/content/v_246873.htm
- 2 <http://en.wikipedia.org/wiki/Firesheep>
- 3 <http://faceniff.ponury.net/>
- 4 <https://www.eff.org/https-everywhere>
- 5 http://vil.nai.com/vil/content/v_501748.htm
- 6 http://vil.nai.com/vil/content/v_522281.htm

Future Threats

Attackers are having good luck going after your private information on smartphones. Although the electric car with a smartphone asks you for permission before enabling its data-sending features, future embedded smartphones may not show the same courtesy.

As more personally identifiable information is controlled by smart hardware, it is the hardware that will become a target for attackers. The Center for Automotive Embedded Systems Security (CAESS), a joint venture of two universities, has demonstrated an attack started by inserting a CD with malicious MP3 files into the sound system: the malware takes over the automobile's engine control unit computer. The team members succeeded in shutting down a running car by controlling the engine, but they could have simply gone after personal data instead.

In the future, attackers will likely spend plenty more effort going after all areas and devices that store your private data.



Jimmy Shah is a mobile security researcher for McAfee. He works in mobile/embedded systems security. If an object is lighter than a car, has a microprocessor, and is likely to be a target, then it's probably Shah's problem. He regularly presents on mobile threat research at computer security conferences.

Social Engineering Eases Real-World Penetration Testing

By David Kennedy



Information security has become something few have ever anticipated and many still struggle to grasp.

Most companies understand to some degree that security is a necessity for protecting brand reputation, intellectual property, financial standing, regulatory requirements, and compliance. Security now plays a much more important role than it ever has; this year alone has proved to be significantly more challenging than past years. Companies are increasing their budgets for protection, purchasing more security technology than ever before, and organizations have dedicated personnel and consultants to safeguard their assets.

Investing in Security Technology

With so much invested in protecting data and boosting security, why do we see more and more breaches occurring with apparently little effort from cybercriminals and hackers? The sad truth is that we can never predict all of our risk and protect against every avenue of attack. Instead, we often immerse ourselves in complex risk equations, formulas, and the most expensive shiny device we can purchase in hopes of solving the question, “How can we

If attackers have trouble penetrating the perimeter, they will naturally try to exploit the weakest link in security—the people.

become secure?” Though it may seem difficult, the answer has been in front of us all along: through hard work and effort. Placing a house on top of a muddy hill without a foundation doesn’t last; neither does purchasing a vast array of preventive technologies without a solid security foundation.

In my work as a “simulated attacker,” or penetration tester, I find it trivial to attack organizations, circumvent their security controls, and gain access to everything they consider sensitive. Companies focus heavily on perimeter security, spending millions of dollars on firewalls and other protective technology, while we generally fly over all of that and gain access without ever touching the perimeter. Some defenders still find the perimeter full of holes and challenging to secure, but we have seen a gradual increase in effective security around Internet-facing systems. If attackers have trouble penetrating the perimeter, they will naturally try to exploit the weakest link in security—the people.

Finding the Easiest Way In

When performing penetration testing, it has never been easier to pick up the phone, impersonate someone of importance, and have the listener do whatever you want. Social engineering is the act of persuading and manipulating one or more persons to surrender information that is not normally available. In most cases, it is easier and more efficient to leverage humans than it is to attack a company’s technology. Think about it for a minute: would an attacker rather fight millions of dollars of investment in detection and prevention or pick up the phone or send a targeted email? The choice for attackers is easy—skip the perimeter attacking, and call or email.

As a penetration tester I know it takes roughly three to four days (sometimes more) of full-on attack to find the one hole that will allow us access to a system. And this works only if we are stealthy enough not to be picked up by alarm mechanisms or the incident-response team. I find that real-world examples work the best. Company A had undergone penetration tests for several years. They were accustomed to the standard SQL-injection attacks and buffer overflows and had a significantly locked-down perimeter. Any web-based, network, or operating system attacks were caught by the security information and event management (SIEM) system and frequently alerted the incident response team. Within a few minutes of starting a penetration test against Company A, the jig was up, and detection had occurred, even with slow port scans. Pretty impressive, right?

Next, instead of choosing the normal attack vectors, my group decided to spend two weeks developing a zero-day assault targeting multiple applications. After two weeks of fuzzing, bypassing Microsoft Windows protection mechanisms, and rigorous testing, the exploit was successful. Within a few minutes, however, detection again occurred as soon as we followed the typical path of hopping onto other systems to further penetrate the network. It was obvious that the direct route was both challenging and time consuming.

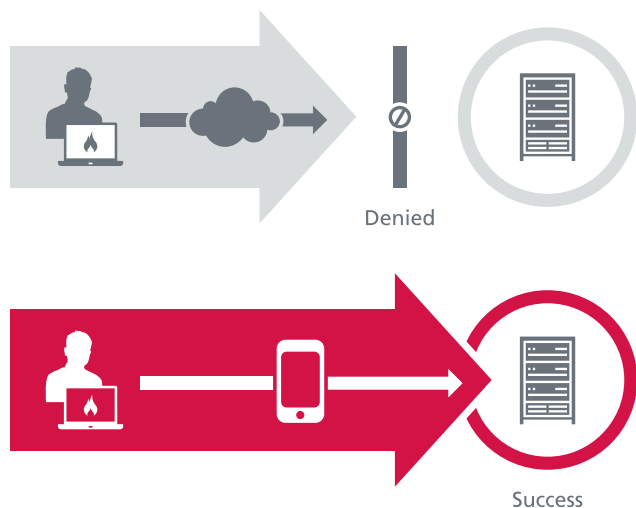


Figure 1. Attackers are happy to follow the path of least resistance when trying to penetrate an organization. Fighting through a firewall and other defenses is more difficult than fooling a person by phone.

Social Engineering

We found that taking the direct route of attacking the organization was not working, so we formulated an alternative method. We spent four hours learning about the organization's hierarchy through open-source intelligence and crafted a specific attack. In social engineering, your attack vector is generally called a pretext—what your scenario will be or the role you will assume to persuade your target. In this instance, the attacker assumed the role of a vice president in the company. We used LinkedIn to identify new employees, who are typically the easiest to persuade, and our attack started to turn into a viable option. After a few calls to numbers found publicly on the Internet, we had identified the numbers of several new employees, thanks to some very helpful people within the organization.

Establishing Trust With Authority

For this pretext to work, attackers must do their homework. They need to understand the nature of the business, the players in the game, the lingo, and anything else that can be learned through passive, public means. The employee targeted in this attack was a new helpdesk team member; these people generally have elevated rights in the organization and regularly deal with calls. When a system is reliant on customer experiences, ticket opening and closing, and system or network performance, such a target is the perfect victim. In our case, the attacker claimed to be a vice president of sales.

"Greetings, Joe," our attacker said. "This is Sam Wallace. I'm vice president of sales and marketing. How are you?"

"I'm doing great," Joe replied. "How can I help you today?"

"I'm experiencing a significant issue. I have a multimillion-dollar bid out to a customer, and I have to be able to access their website. For some reason, the latest update seems to have broken the ability to run their site properly. I'm really in a time crunch to win this bid. If you can help me out, I will be forever grateful and will let your management know how well you did!"

"What website are you trying to hit?"

"For some reason, when I go to <http://xxxxxx.com> and I run their proposal applet, it doesn't load right. Can you try it on your end?"

"Sure. One second, sir. I'm testing it right now."



Figure 2. The Social-Engineer Toolkit can impersonate a Java applet and fool a victim into downloading malware.



Figure 3. In this image, the Social-Engineer Toolkit interactive shell has full access to the victim's machine. The command prompt demonstrates an established connection.

The savvy attacker previously did some homework and registered a domain name similar to a customer of the organization. The attacker had already fired up the Social-Engineer Toolkit, a widely popular penetration testing tool for social engineering, and set up the attack. The Social-Engineer Toolkit specifically performs attacks relating to social engineering and relies heavily on human interaction for success. In this instance, the attacker leveraged the “Java Applet” attack vector.

When Joe, our victim, clicked “Run” on the malicious Java applet, his machine was compromised. “It appears to be loading fine for me,” Joe said. “Can you test it out for me and see what you are experiencing?”

“Sure, one minute. Whoa! I’m not sure what you did, but it’s working now. Thank you so much. I will let your management know that you are a rock star and just saved the company millions of dollars!”

As the attacker hung up, Joe was ecstatic. He had been on the job for only a few weeks and already had a large win with an executive. Unknown to him, unfortunately, he had been fully compromised by the attacker.

At this point in the penetration test, we had full access to Joe’s computer. The attacker leveraged the information obtained on the system, for example, local administrative rights, valid Kerberos tokens, and anything else useful, and further penetrated the network. As penetration testers, our goal is typically to identify how to inflict the most damage to an organization. This could occur through theft of highly sensitive intellectual property, access to sensitive systems, or performing massive brand destruction leading to big losses in revenue. In this attack, we gained full access to the company’s most sensitive secrets. Had those reached competitors or the Internet, Company A would have suffered irreparable damage.

As attacks evolve and the threat landscape continues to grow, information security needs to take a fluid and flexible approach to minimize the damage of a breach.

Hauling in the Anchor

We used this scenario in a real penetration test of a Fortune 1000 company. The particular social engineering attack we leveraged is one of my favorites, called anchoring. Anchoring places an “anchor” in the head of the victim, and then the attacker pulls in the anchor. In this test, the attacker impersonated someone of authority and placed the anchor by stating that there was a problem with a software update and millions of dollars were on the line. Immediately the victim thought that someone in authority had a significant problem. The attacker praised the victim and said that if he could resolve the issue, he would be rewarded. At that point, trust had been established in the mind of the victim and the anchor was ready to be pulled. The attacker needed only to ask for help and Joe was ready to comply. The anchor attack was successfully completed.

Anchoring is just one method for social engineering an exploit with an individual and is incredibly hard to protect against. During this attack, several alarms should have gone off in the employee’s mind. He should have verified the identity of the caller, yet he did not do so because the victim wanted to please. He was further hampered by the lack of formalized training on user awareness and the dangers of social engineering.

Two Years Later

Let’s leap forward two years. The results of our penetration test inspired Company A to embark on a massive user-awareness campaign. They dedicated resources to arming employees with indicators and the knowledge to successfully detect these types of attacks. After the employee training effort, we launched a similar attack against a different business unit, this time the sales force. The attack was flagged as suspicious and the incident-response team was notified. Although this rapid response won’t happen 100 percent of the time, understanding how social engineering attacks work and how protective mechanisms can be put in place—both from a technological and human perspective—is priceless.

As attacks evolve and the threat landscape continues to grow, information security needs to take a fluid and flexible approach to minimize the damage of a breach. Attackers need to be right only one time to successfully damage an organization; thus how we respond before, during, and after an attack is vital. If a security program is robust and flexible enough to detect the latest exploit, SQL injection, or employee attack, that program is well positioned to protect the organization.

Conclusion

Information security is a new field. The industry still has a lot to learn from its mistakes, but one thing is clear. If we continue on our current path, we will continue to experience a number of security breaches and loss of revenue if we can’t handle targeted attacks. Companies that base their information security program on compliance have a long road to travel. In my experience, compliance is an awesome driver for funding and awareness around security, but the foundational security that a program requires cannot come through compliance-related efforts or the latest technological toy that promises to protect you against attacks.

The game is changing. If we move forward in building our security programs to handle these types of attacks, then we will have a chance of sustainability. If we move away from hard work and effort and rely on automated tools and compliance standards, then the industry will have a much more challenging time.



David Kennedy is chief information security officer at Diebold Inc. and creator of the Social-Engineer Toolkit, Fast-Track, and other open-source tools. He is on the Back|Track and Exploit-Database development team and is a core member of the Social-Engineer podcast and framework. Kennedy has presented at security conferences Black Hat, DEF CON, ShmooCon, B-Sides, and others. He is one of the cofounders of DerbyCon, an information security conference in Louisville, Kentucky, and is a coauthor of *Metasploit: A Penetration Testers Guide*, from No Starch Press. Kennedy is supported by his wife, children, and his MacBook Pro quad-core with solid-state drive and eight GB of RAM.

The Cloud Is Changing Everything

By Scott Chasin



Each year, I make a pilgrimage to the Consumer Electronics Show (CES) in Las Vegas. Dubbed the World's Largest Technology Show, the event attracts tens of thousands of geeks and non-geeks from around the world.

Although attendance was down this year over previous years, the show still attracted more than 125,000 people, and it didn't disappoint.

This year, I watched robots ride bicycles. I sat back in amazement as I viewed a 3D television without the need for the goofy glasses. I was impressed by the demo of a "smart" dishwasher that can send email notices when the dishes are clean. As it turns out, that's important information to some people.

Amid the hype and excitement of all these futuristic gadgets, a couple of broad themes emerged. Foremost was the explosion of mobile devices. There were no fewer than 80 product announcements of tablet computers, each of which promised to do more for less money than the world-famous Apple iPad. The second big trend was the hype around Verizon's next-generation 4G network and the ability to connect to the Internet "cloud." That message was everywhere.

Later, I was reflecting on everything I saw at CES while trying to log onto our company's network through a secure VPN connection. After several failed attempts, I simply gave up. And that got me thinking. When it comes to technology, how is it possible that we can be such powerful consumers, yet such lame employees?

The simple fact is that cloud-based computing and the consum-erization of IT are changing our world. As IT professionals or business owners, we simply have to accept this fact. The network perimeter as we've known it is gone. It's not in the process of going away; it's simply gone. We need to either adapt and embrace these new technologies or accept that our competitors will. Some of them already have.

Of course, relying on the cloud is not without risks. Hackers have already taken notice of the interest and demand for cloud-based computing. That's where they'll focus next—and that should be no real surprise. After all, the bad guys follow the information. Where there's information to be stolen, there's money to be made. So as more and more information and applications migrate to the cloud, so too will the threats.

Leveraged correctly, the cloud both helps reduce your security costs and can actually increase your overall security posture and protection.

Create or Destroy

Like all new technologies, the cloud presents both risks and rewards. As US President Barack Obama said in a recent speech, “The very technologies that empower us to create and to build also empower those who would disrupt and destroy.” This is certainly true with the cloud. In fact, although it is difficult to prove, many of my security colleagues speculate that the world’s largest user of cloud computing isn’t Google, Microsoft, or even Amazon. Rather, it’s the Conficker botnet.

Even so, those who ponder the potential risks of the cloud often fail to recognize that the cloud is not only good at scaling a business’s computing power, it’s also a highly efficient way to scale security and protection for a business. Leveraged correctly, the cloud both helps reduce your security costs and can actually increase your overall security posture and protection.

Cloud Security

The trouble with the cloud is that it’s an abstract idea with multiple definitions. To simplify things and help people understand how they can leverage the cloud for security, we break things down into three basic benefits:

- Reduce IT costs by using security solutions delivered from the cloud using the software-as-a-service model (SaaS)
- Improve threat protection by using security solutions that leverage global threat intelligence networks and technologies to collect and correlate real-time threat data in the cloud
- Increase the protection of data and applications stored in public or private clouds with security solutions designed specifically for the cloud

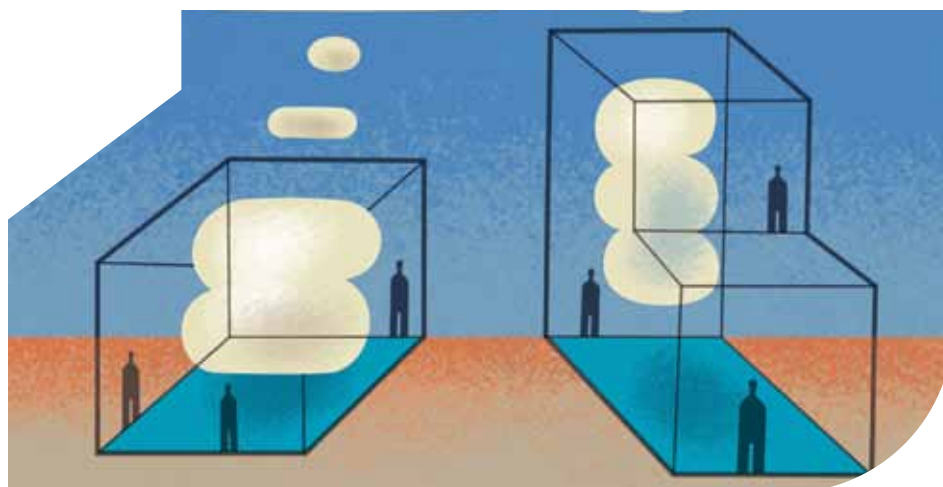
Although these three steps don’t cover all aspects of cloud security, this simple framework provides a good starting point for figuring out how to leverage the cloud for security. In addition, these often provide a systematic path for integrating cloud security into your current security infrastructure.

Security from the Cloud: Reducing Costs, Increasing Protection with SaaS

SaaS security solutions are probably the most common and easiest to implement because they’ve been around for years. Rather than buying and managing security hardware or software, businesses can subscribe to pay-as-you-go SaaS security for everything from email and web to endpoint protection. The most obvious benefit of SaaS security is the elimination of costly capital expenditures and the ongoing management costs. The savings can be significant. A recent Tolly Group study revealed that companies using a SaaS security solution versus a traditional hardware or software security solution saved as much as 50 percent. Moreover, security SaaS is generally considered an operating expense rather than a capital expenditure, making budgeting much easier and more predictable.

Beyond the cost-saving benefits, SaaS security solutions can also be more effective. A 2010 Aberdeen study showed that companies using SaaS security for email protection reported 47 percent fewer incidents of spam during a 12-month span. Businesses using SaaS security for web protection also reported 58 percent fewer malware incidents during the same period. Because hardware and SaaS security solutions generally use the same basic technologies, the difference in protection is attributed to the managed nature of SaaS solutions. The vendor is responsible for managing updates and making sure the SaaS solution is functioning at peak performance around the clock.

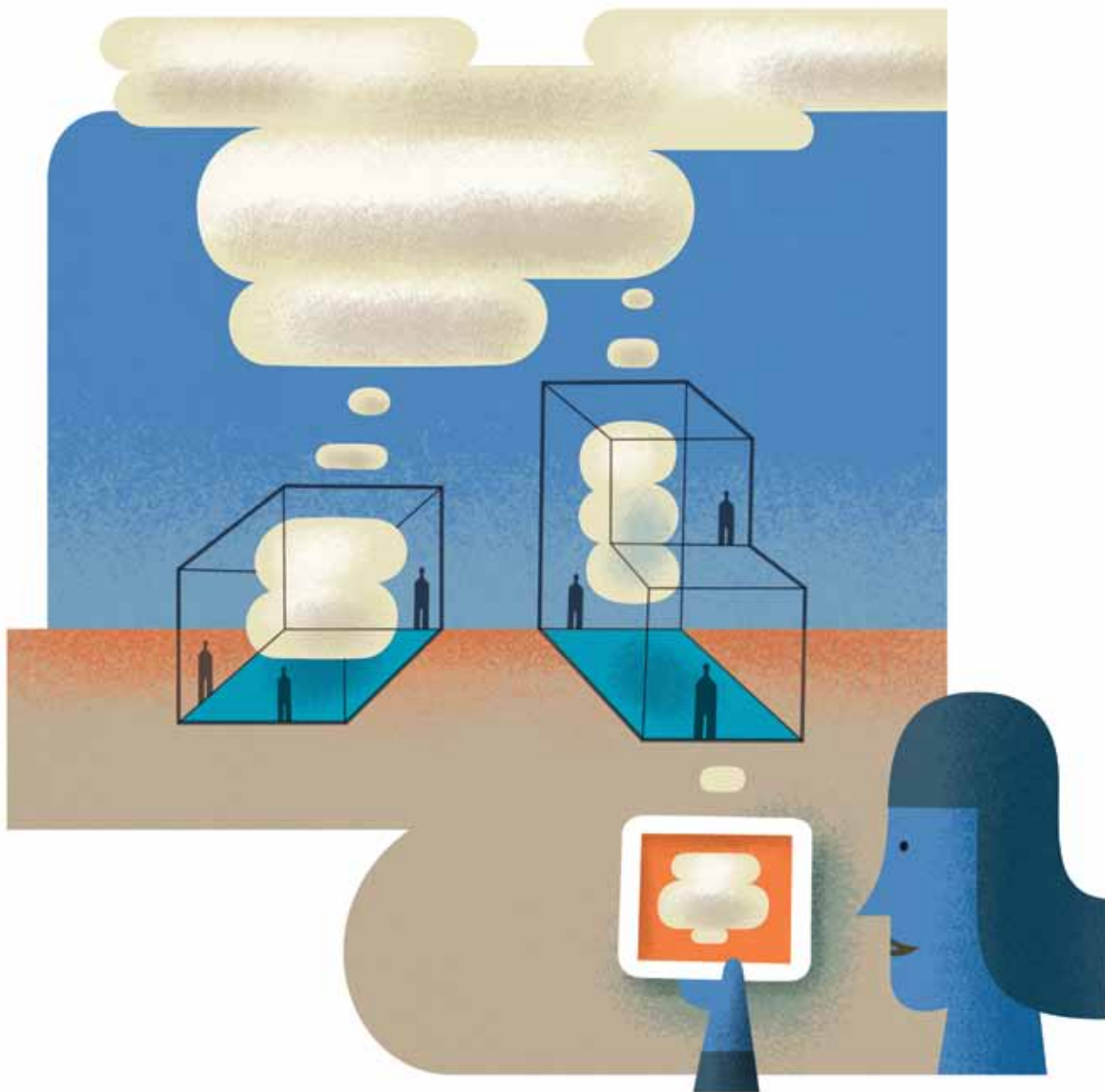
The biggest challenge with SaaS security is picking the right vendor. Many vendors offer a single stand-alone SaaS security solution. A few others offer one or two services together. The ideal solution for customers is a broad range of integrated SaaS security, ranging from email and web to mobile and endpoint solutions. Expertise also counts. Service-level agreements (SLAs) are great, but in the end, they’re just promises—not guarantees. They should never substitute for checking a vendor’s history, reputation, and track record.



Security in the Cloud: Leveraging the Cloud for Better and Faster Threat Protection

Another easy way scale your protection is by using solutions that leverage the cloud to collect and correlate threat data in the cloud. In the end, security solutions can stop only what they are looking to stop. The more security threats a vendor can see and identify, the more likely that vendor can prevent them from infecting your network.

Speed also counts. In the past, security vendors typically took anywhere from 48 to 72 hours to identify a new threat, write the signature file to protect against it, and distribute that file to their customers. It's more effective to leverage the cloud to compress time to protection by turning customers' endpoints—firewalls, laptops, mobile devices, and others—into listening posts. Each of these devices can report unusual threat activity back to the vendor to analyze threats in real time. What used to take 48 to 72 hours can now take place in less than 15 minutes. In some cases, the visibility the cloud provides can allow a vendor to predict large-scale threats before they happen. The Operation Aurora attack, which targeted high-profile global enterprises and governments last year, was thwarted by one vendor before it was launched. As a result, none of its customers was affected by the operation.



Security for the Cloud: Increasing Protection of Data and Applications

Despite the enormous interest in cloud computing, security remains a top concern cited by customers. According to IDC, nearly nine out of 10 businesses cite “security concerns” as their primary obstacle for adoption. This is no surprise, as storing information or running applications in the cloud is similar to walking into a room with the lights turned off. Your visibility into what is going on with your data and applications—or who has access to it—is greatly reduced, if not eliminated altogether.

The key is transparency and validation. The challenge, however, is how to get it. Most cloud-computing vendors assure customers that they are doing what is necessary to protect your data. After all, they argue, “Our reputation is at stake if your data is lost or compromised.” That’s a logical claim, but it does little to reassure even the most liberal customer.

Cloud vendors are also quick to point to their annual security certifications and audits. Although these are a good step, they rarely go far enough. As everyone knows, an audit or certification is a snapshot in time. Just because a cloud vendor passed an audit two or six months ago doesn’t mean it is not vulnerable to threats today.

Security vendors need to help bridge the gap between cloud providers and enterprises by enabling more transparency. Global organizations such as the Cloud Security Alliance are working to define industry standards and best practices for cloud security.¹

Get Started

Sometimes the most difficult part of leveraging the cloud for security is getting started. Luckily, cloud security is not an all-or-nothing proposition. Many organizations start with something simple, such as email or web SaaS security solutions. This allows them to test the waters and overcome any of the natural—and healthy—skepticism that goes with using new technologies.

While we often think of the cloud as the ideal way to scale a business, I encourage you to consider the cloud as a way of scaling your protection, too. Your competitors certainly are.



Scott Chasin, McAfee chief technology officer, content and cloud, is widely recognized as a leading visionary in the cloud security industry, having pioneered the development and marketing of several SaaS-based messaging, collaboration, and security-focused technologies. He launched the first web-based email consumer service and delivered the first IP-based commercially hosted messaging service supporting thousands of businesses and millions of users worldwide. Chasin also created and moderated the first full-disclosure security discussion list, Bugtraq, and was a key contributor to the creation of the first open-source, one-time password system, S/Key. A frequent speaker at industry conferences and tradeshows, he blogs regularly at www.mcafee.com/cloud.

ENDNOTES

¹ <https://cloudsecurityalliance.org/>

SQL-Injection Attacks Too Easily Threaten Data

By Vadim Pogulievsky



Several recent stories of security breaches were so big that it would have been difficult not to notice to them.

Hackers made off with a database containing names, email addresses, and Vehicle Identification Numbers [the unique ID for cars] of 2.2 million Honda customers following an attack on an unnamed third-party marketing outfit.

—The Register¹

Epsilon, which sends 40 billion emails annually on behalf of more than 2,500 clients, said a subset of its clients' customer information was compromised in the data breach.

—ABC News²

Hackers say they breached the website security of computer-maker Acer and made off with data for 40,000 of its customers.

—The Register³

In addition to these three, other security breaches in recent months involved RSA, Barracuda Networks, MySQL, HBGary, Citigroup, and a series of attacks on Sony. We could also mention the LizaMoon mass SQL injection, although this breach was not a targeted attack on a single big company or vendor; yet it was probably the most significant security issue in April.⁴ The attack code, injected into legitimate web pages, redirected browsers to a malicious server that tricked users into downloading and installing “fake anti-virus” malware, also known as rogue anti-virus software. One report claimed more than one million sites were compromised; that figure appeared to position LizaMoon as a SQL-injection attack of the first order, similar to Gumbler a couple of years ago. However, subsequent research has shown that LizaMoon’s infection rate was much lower than originally estimated.

What is the common thread in all these attacks? Excluding some hacktivism cases, they all target data. In our “data driven” world, this shouldn’t be a big surprise because data is the core of business systems. From the security perspective, this suggests that data storage systems should be the best protected resources in any organization.

But are they really? Unfortunately, the answer appears to be “no.”

Databases are subject to various attacks, from both internal and external sources. The biggest database vendors fix dozens of new vulnerabilities every year. (Here we have to say a good word about Microsoft, whose SQL Server 2008 is one of most vulnerability-free systems.)

There may be many reasons for the incredible popularity of the SQL-injection technique, but the two most significant are probably that the overwhelming majority of websites are vulnerable to this type of attack and that SQL injection is relatively simple to implement.

The most common problem related to databases is the SQL-injection attack, a technique in which malicious SQL code is injected into user input and is processed in a database layer of the application. Not a database vulnerability per se, SQL injection is the most abused technique for data-targeting attacks. SQL injection, which was first described in the middle 1990s, is still considered the most critical web application security risk by the Open Web Application Security Project Top 10.⁵

In the preceding data-theft cases, we can assume that in the majority of incidents SQL injection was either the only or the chief technique in the attack. There may be many reasons for the incredible popularity of the SQL-injection technique, but the two most significant are probably that the overwhelming majority of websites are vulnerable to this type of attack and that SQL injection is relatively simple to implement.

Unfortunately, it's not just stand-alone websites that are vulnerable to SQL injection—even professional web applications and platforms suffer from the same problem. This weakness allows hackers to successfully exploit a vulnerability multiple times for different attacks.

Looking at the web vulnerabilities section on Exploit-DB.com, we can see that close to half of all vulnerabilities discovered in one week were SQL injections. Some of them affect fairly popular web applications. Unfortunately, neither this particular week nor these particular sites are out of the ordinary; the huge number of newly discovered SQL-injection vulnerabilities is a well-established trend. SQL-injection vulnerabilities are so common they probably compete only with cross-site scripting for the distinction of most popular web attack.

The second reason for the great popularity of SQL injection is the relative simplicity of implementing an attack. In many cases, an attacker requires only primitive knowledge of SQL to be successful. A huge amount of training material is available online that definitely multiplies the effect. Finally, if an attacker doesn't have sufficient knowledge to launch an attack, he or she can always find somebody who will offer support.

In addition to finding easy advice, those who don't want to bother with technical nonsense can choose among commercial and free automated SQL-injection tools. Automated tools such as SQLMap, Netsparker, and many others are developed to perform penetration tests and self-audits. These tools are very productive for scanning large amounts of web pages. However, they become a dangerous weapon in the hands of script kiddies, who can carry out dangerous attacks with little technical knowledge.

Date	ID	Description	Pwn	Type	Author
2011-05-31	10000	WordPress 2.9.2 Remote File Inclusion Vulnerability	75	File Inclusion	0x00000000
2011-05-31	10001	Drupal 6.19.0 Remote File Inclusion Vulnerability	70	File Inclusion	0x00000000
2011-05-31	10002	Drupal 6.19.0 Remote File Inclusion Vulnerability	70	File Inclusion	0x00000000
2011-05-31	10003	Drupal 6.19.0 Remote File Inclusion Vulnerability	70	File Inclusion	0x00000000
2011-05-31	10004	Drupal 6.19.0 Remote File Inclusion Vulnerability	70	File Inclusion	0x00000000
2011-05-31	10005	Drupal 6.19.0 Remote File Inclusion Vulnerability	70	File Inclusion	0x00000000
2011-05-31	10006	Drupal 6.19.0 Remote File Inclusion Vulnerability	70	File Inclusion	0x00000000
2011-05-31	10007	Drupal 6.19.0 Remote File Inclusion Vulnerability	70	File Inclusion	0x00000000
2011-05-31	10008	Drupal 6.19.0 Remote File Inclusion Vulnerability	70	File Inclusion	0x00000000
2011-05-31	10009	Drupal 6.19.0 Remote File Inclusion Vulnerability	70	File Inclusion	0x00000000
2011-05-31	10010	Drupal 6.19.0 Remote File Inclusion Vulnerability	70	File Inclusion	0x00000000
2011-05-31	10011	Drupal 6.19.0 Remote File Inclusion Vulnerability	70	File Inclusion	0x00000000
2011-05-31	10012	Drupal 6.19.0 Remote File Inclusion Vulnerability	70	File Inclusion	0x00000000
2011-05-31	10013	Drupal 6.19.0 Remote File Inclusion Vulnerability	70	File Inclusion	0x00000000
2011-05-31	10014	Drupal 6.19.0 Remote File Inclusion Vulnerability	70	File Inclusion	0x00000000
2011-05-31	10015	Drupal 6.19.0 Remote File Inclusion Vulnerability	70	File Inclusion	0x00000000
2011-05-31	10016	Drupal 6.19.0 Remote File Inclusion Vulnerability	70	File Inclusion	0x00000000
2011-05-31	10017	Drupal 6.19.0 Remote File Inclusion Vulnerability	70	File Inclusion	0x00000000
2011-05-31	10018	Drupal 6.19.0 Remote File Inclusion Vulnerability	70	File Inclusion	0x00000000
2011-05-31	10019	Drupal 6.19.0 Remote File Inclusion Vulnerability	70	File Inclusion	0x00000000
2011-05-31	10020	Drupal 6.19.0 Remote File Inclusion Vulnerability	70	File Inclusion	0x00000000

Figure 1. Looking at Exploit-DB.com confirms that SQL-injection attacks are very common.

```

to all

this site database is POSTGRE

http://[redacted]/?id=22&id=8

I know this order only

--and 1=CAST(current_user()CHR(58)||current_database()CHR(58)||version()CHR(58)||123
as int)

look is

http://[redacted]/?id=22&id=8--and
1=CAST(current_user()CHR(58)||current_database()CHR(58)||version()CHR(58)||123 as int)

Result is:

PostgreSQL 8.3.15 on x86_64-pc-linux-gnu, compiled by GCC cc (GCC) 4.2.4 (Ubuntu 4.2.4-
1ubuntu4)

How to inject this kind of sites.

help site:

Options: Reply • Quote

Re: POSTGRE SQL Injection pitz help
Posted by: 0x00000000
Date: May 30, 2011 12:09PM

Look:
http://[redacted]/?id=44&id=7 and 1=0 UNION SELECT
null,version(),null,null,null,null,null,-- --
  
```

Figure 2. Even inexperienced attackers can find advice in forums.

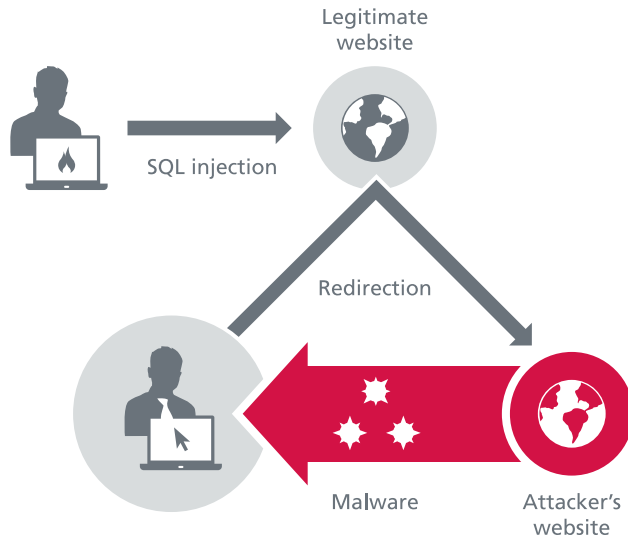


Figure 3. A mass SQL-injection attack places malware on the systems of visitors who come to a legitimate website.

SQL-injection attacks differ depending on the database of a particular website. This variation increases the complexity of the SQL-injection attacks because attackers must be aware of SQL flavors, particularly security restrictions, default configurations, and many other parameters of each database. However, these challenges become important only at a relatively late stage, when the attack might already have been discovered. At this point, an attack has traveled a fairly long way to the target.

As a popular hacker's technique, SQL-injection attacks have different goals. The most straightforward attack steals data. This assault occurs mostly in targeted attacks on web systems of companies and organizations. As *targeted* implies, these attacks are unique, aiming for specific web systems.

From the specific, we move to broad-scale attacks. The latest example of this kind is LizaMoon. The goal of this mass SQL injection is conceptually different from targeted breaches. In targeted attacks, the primary victim is an organization, but in mass attacks, the victims are visitors of hacked websites. The concept works as follows: hackers, using SQL injection, insert malicious code along with a website's normal content. This code either directly installs malware on victims' machines or redirects visitors' browsers to external malicious pages.

Figure 3 shows a victim's machine being exploited as a result of visiting a legitimate website, perhaps the victim's everyday news portal. Although the primary victims of this kind of attack are the website's visitors, the site and business also suffer from the attack because the loss of reputation is a very significant issue, especially considering the number of and easy access to competitors. To launch a mass SQL injection, a hacker usually seeks a common denominator for all sites under attack to automate the exploitation process. Server-side web applications or content management systems are the common targets.

The data security world probably is the most volatile of information technology domains, as attackers and defenders periodically gain the upper hand. Every new attack technique causes a reaction from security companies, which invest in new technologies and products. And every new security technique causes the bad guys to puzzle over how to bypass the technology.

Unfortunately, this back and forth isn't applicable to SQL-injection attacks. We have seen no significant steps forward on either side for a long time. The current situation is not encouraging. The bad guys do not need new inventions because the well-tested SQL-injection techniques work flawlessly for the majority of cases. The LizaMoon attack, for example, successfully exploited more than one million web pages. (Fortunately many of the LizaMoon exploits did not reach their final goal because of the congenital disadvantage of mass infections: the lack of the "personal touch.") As an automated process, mass injections rely on a general approach, missing the particular nuances that a targeted attack would include.) Asprox, a similar attack in 2010, succeeded in injecting SQL code on a few thousand ASP-powered websites.

However, the ease of creating attacks doesn't mean that absolutely nothing happens in the SQL-injection world. Most big security conferences include presentations related to the issue. In most cases, security professionals present their investigations related to advanced injection techniques. We've recently seen several hot topics published or presented:

- Performance improvements in blind SQL injections allow attackers to minimize the amount of time and requests to successfully extract data from the victim's database
- New methods using error-based SQL injections allow attackers to extract data based only on error messages presented at the web page after specially crafted SQL requests
- Out-of-band connections through different protocols allow attackers to send data out of a database to the attacker's server using the victim's database communication abilities

The Next Generation

The recent sequence of database breaches will increase the awareness of database security, and more and more companies will start to invest significant resources in database protection products. This trend will slowly but surely change the status quo. The attackers' response is predictable: they will improve attack obfuscation techniques to bypass security products. We have seen them do this before, with malware-executable obfuscation techniques and with JavaScript and PDF drive-by download exploits.

We don't expect to solve the SQL-injection problem soon because the roots of the issue lie in insecure development procedures. Even today many developers are not aware of security issues in general and of combating SQL injection in particular. The common opinion is that SQL injection is the exclusive problem of web developers. However, as practice shows, SQL injection can exist in any application layer: web, business logic, and databases themselves. In all cases, vulnerabilities could be prevented by proper and secure code-development standards. Although these standards are well known, many developers still find it difficult to follow them.

First, developers should use parameterized statements as a secure alternative to commonly used dynamic SQL, which is the essence of every SQL injection. Second, proper input validation is a common security problem—abused not only in injections of all kinds, but also by many other exploits, for example, buffer overflows. Several further mechanisms are available to improve security or minimize the consequences of attacks. Considering the global scale, the current results are lamentable.

No Cause for Despair

In spite of the advantages attackers enjoy, that doesn't mean that nothing can be done. The good news is that in each particular case, with a corresponding effort and knowledge, the problem is far from unsolvable. The security concept is similar to a steeplechase: the more obstacles placed in the attacker's way, the lower the chance that a system will be hacked. By following the secure development concepts mentioned before, plus setting proper database configurations and restrictions, performing periodic audits, and implementing protecting and monitoring security products, we can create effective defense in depth—the security “obstacles” that must be used in a properly secured website.

Remember, in 99 percent of attacks, hackers look for the low-hanging fruit. Preparing and carrying out an attack is a time- and effort-consuming task. Attackers will most likely pass and assault other systems if they cannot find a good attack vector on the current one.

“In the zombie apocalypse, you won't have to be faster than the zombies, you'll just have to be faster than your friends.” Every joke has some truth in it, no?



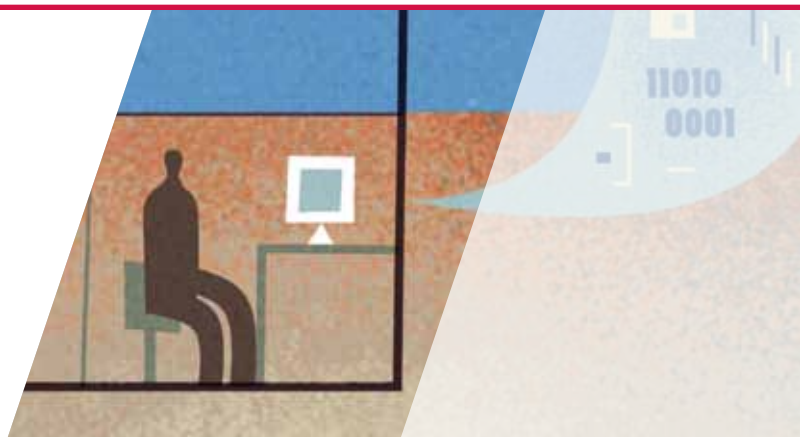
Vadim Pogulievsky is a security research manager for McAfee Labs in Israel. His current research focuses on database security, but his interests also stray to web security, vulnerabilities research, and exploits development. Prior to joining McAfee, Pogulievsky was a manager of the security research team at M86 Security and at Finjan.

ENDNOTES

- 1 http://www.theregister.co.uk/2010/12/31/honda_data_breach/
- 2 <http://abcnews.go.com/Technology/epsilon-email-breach/story?id=13291589>
- 3 http://www.theregister.co.uk/2011/06/03/acer_customer_data/
- 4 <http://www.reuters.com/article/2011/04/01/hackers-idUSN0116927520110401>
- 5 https://www.owasp.org/index.php/SQL_Injection

Can Studying the Past Help Secure the Future?

By David Marcus



I have long been a student of two things: history and the mystery novels of Agatha Christie. I study history because very few things are really new or without precedent.

Examining battles, empires, civilizations, and events provides insight into some of the greatest mistakes and successes of the past, providing us the opportunity to learn, adjust strategies, and even avoid common pitfalls by analyzing how others dealt with similar situations. History is a great teacher for those who take the time to study it.

Use Those “Little Grey Cells”

Why Dame Agatha? Aside from enjoying a good murder mystery, I like two main things about her approach to crime solving: method and observation. Through her protagonist Hercule Poirot, Dame Agatha taught me order and method. Look at the facts, and then stop and think. As the great Belgian detective often said: “We must use our Little Grey Cells!” In other words, we must think. Poirot was able to solve mysteries others could not simply by assembling all the facts and then taking the time to think, process, and ask questions. Christie’s Miss Jane Marple taught me several other skills: to keenly observe behavior and look for commonalities. Aunt Jane lived in a “quiet” English village, but she often said, “There is more evil in an English village. You just have to know where to look.” She was a sharp observer of human

behavior. She could also find analogies and similarities in the events and behaviors of the past and use them to see similarities in current events. This analytical ability led her to insights others could never achieve. Often she would see a face or event that reminded her of people or events in the past. The similarities allowed her to guess how similar people or events would act or unfold in the future. Like Poirot, Marple was also right-hand-side-of-the-Bell-curve smart, often described as having “a mind like a meat cleaver.” They both observed, asked questions, and thought deeply. When they arrived at a conclusion, they didn’t let it go.

Now let’s take those same skills and apply them to recent information security events, specifically hacktivism.

Hacktivism Is the Message

Hacktivism, a portmanteau of hacking and activism, refers to using the skills of hacking to achieve an activist’s goal. From a historical perspective, this is not new. The phrase was arguably coined in 1995 (though some claim that it was first used in 1994 by Cult of the Dead Cow [cDc] member Omega) in an *InfoNation* article by Jason Sack,¹ but instances of hacktivism can be traced back to as early as 1989, when the anti-nuclear WANK worm targeted the US Department of Energy-, HEPNET-, and SPAN (NASA)-connected VMS machines.² Groups like Cult of the Dead Cow (and its cDc Communications), formed in 1984 and still active today, were not originally formed for hacktivism, though cDc did expand its “charter” to include the independent group Hacktivismo in 1999. Hacktivismo was dedicated to the creation of anti-censorship technology in furtherance of human rights on the Internet. The group’s beliefs are described fully in “The Hacktivismo Declaration,”

written by legendary hacktivist Oxblood Ruffin. The article seeks to apply the Universal Declaration of Human Rights³ and the International Covenant on Civil and Political Rights⁴ to the Internet.⁵ All are worthy of full reads, as they are compelling documents. We have seen numerous other examples during the past years, but it is the recent activities of LulzSec and Anonymous that have garnered a great deal of media attention. These groups, though their stated goals and agendas vary, have pushed corporate, enterprise, and user information security issues and data privacy issues to the forefront of world news. Now let's step back and ask some questions.

What Inspires Hacktivists?

Who are the hacktivists, and why are they doing this? There are many active hacktivist groups and individuals. The group Anonymous champions the WikiLeaks cause and stands against corporate and government abuses of power. LulzSec is very different, being in this just for the "lulz" (LOLs, or the laughs), as they claim. There are also individual players such as The Jester (th3j35t3r) who target jihadist, anti-American, and anti-free speech/hate organizations (like the Westboro Baptist Church). The actual number of hacktivists is in doubt, and their levels of organization are unknown. Occasionally they even target each other! There are also movements and operations within hacktivism, such as #AntiSec, an operation started by LulzSec with the goals of embarrassing and combating information security, intelligence, and government agencies.

Those who feel they can stop or end hacktivist activities show that they do not understand the nature of activism. Hacktivism is simply a different means to an end—that of delivering a message or serving as an agent of change.

Many of the players in the hacktivist movement are unknown. Some of the motivating factors seem to be a feeling of powerlessness and a great sense of injustice on the part of oppressive governments, agencies, and corporations. Hacktivism gives a sense of power and voice to these perceived rightful indignations. Many hacktivists might even claim they want to change the world. I see many of the sentiments of cDc Communications in the current works of Anonymous, LulzSec, and other hacktivists. Much of the over-sensationalized, alleged in-fighting of these groups originated in the overblown "Great Hacker War" of 1990 to 1991—and much of that buzz was media generated as well.

Those who feel they can stop or end hacktivist activities show that they do not understand the nature of activism. Hacktivism is simply a different means to an end—that of delivering a message or serving as an agent of change.

Well-Chosen Targets

Why are hacktivists so successful? These groups and movements have shown a remarkable agility. They pick their targets well. They have demonstrated that they can get long-term access to servers and data across multiple networks and extrude the data they want. They operate with apparently very little concern for identification or capture and with seemingly little, if any, opposition. Many of the techniques and attacks they use in these operations are not sophisticated or advanced. This is a point we need to examine.

Why are well-known vulnerabilities and weaknesses still so prevalent today? Sure, code mistakes will always happen, but we have methodologies and processes for safeguards such as server lockdowns, password strategies, and database hardening. Maybe the security industry has not educated people in the basics of architecture and deployment. Perhaps we still place too much emphasis on security as a device or add-on. If this is the reason, then perhaps many of today's compromises are partially our own fault. It's time to simplify and return to the basics of security—and those basics are not product related.

I find myself wondering if our industry has simply evolved into uphill boulder pushing, like the endless, unsuccessful travails of Sisyphus. We now call spear-phishing exploits against executives "advanced persistent threats." We're simply changing the name on the boulder pushed up the same hill, and that hill is called security product. I am waiting for the first vendor to market an "anti-LulzSec" or "anti-hacktivism" device.

How can we break this recurring loop in information security? We need to look closely at the events and their outcomes, and then use what we learn to our operational advantage. Replicate the means that LulzSec, Anonymous, and other hacktivist groups use before you get targeted. If that sounds like a penetration test, it should. If you are in a field related to the targets of any of these recent attacks, then you should assume yourself to be at least equally vulnerable and take steps.

Hacktivism, although it contains both constructive and destructive components, is really about the message rather than the method. Let's take advantage of the lessons that hacktivism offers us by closely observing events both current and historic and then by applying order and method. If you are targeted, ask yourself why. Take the time to test thoroughly. Take the time to remediate thoroughly. Take the time to return to security basics that don't rely on products.



David Marcus is director of security research and communications for McAfee Labs. He brings the results of the extensive security research conducted by the labs to McAfee customers and the greater security community. Marcus' responsibilities include public relations, media, and thought leadership; serving as blogmaster for the "McAfee Labs Security Blog"; and co-hosting "AudioParasitics — The Official Podcast of McAfee Labs." He also manages all publications from McAfee Labs, including the *McAfee Security Journal*.

ENDNOTES

- 1 <http://en.wikipedia.org/wiki/Hacktivism#Hacktivism>
- 2 Assange, Julian. "The Curious Origins of Political Hacktivism," CounterPunch, Nov. 25, 2006. <http://www.counterpunch.org/assange11252006.html>
- 3 http://en.wikipedia.org/wiki/Universal_Declaration_of_Human_Rights
- 4 http://en.wikipedia.org/wiki/International_Covenant_on_Civil_and_Political_Rights
- 5 Ruffin, Oxblood. "The Hacktivism Declaration: International bookburning in progress," July 4, 2001. http://www.cultdeadcow.com/cDc_files/declaration.html



McAfee
2821 Mission College Boulevard
Santa Clara, CA 95054
USA
888 847 8766
www.mcafee.com

McAfee, the McAfee logo, and McAfee Labs are registered trademarks or trademarks of McAfee, Inc. or its subsidiaries in the United States and other countries. Other marks and brands may be claimed as the property of others. The product plans, specifications, and descriptions herein are provided only for information and are subject to change without notice. They are provided without warranty of any kind, expressed or implied. Copyright © 2011 McAfee, Inc.
31300_sec-jrnl-sum11_0911