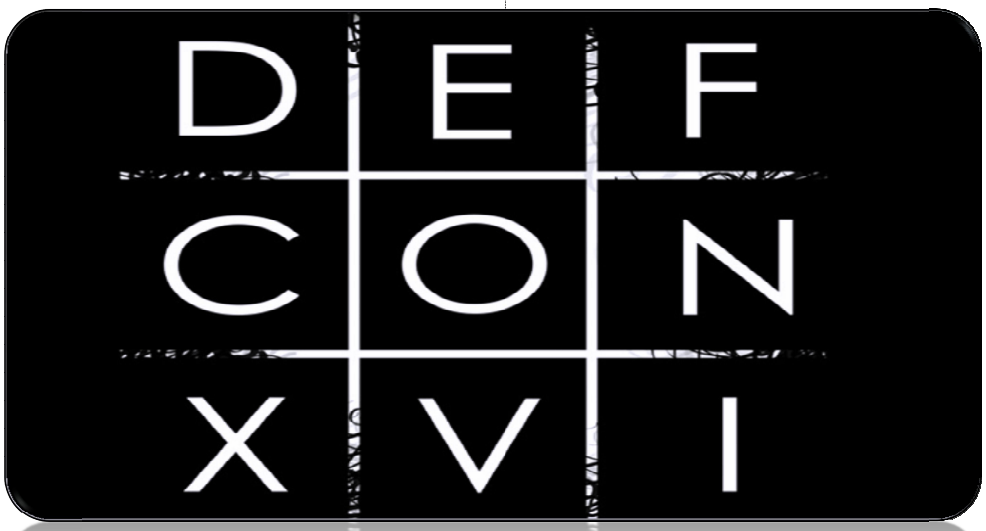
Satan is on my Friends List : *Attacking Social Networks*



:: DefCon Sixteen (FTMFW) ::

Shawn Moyer
and
Nathan Hamiel

"Has what we've learned about writing software the last 20 years been expressed in the design of Web 2.0? Of course not! It can't even be said to have a 'design.' If showing people what vulnerabilities can do were going to somehow encourage software developers to be more careful about programming, Web 2.0 would not be happening.

Trust model? What's that? The so-called vulnerability 'researchers' are already sharpening their knives for the coming feast."

:: Marcus Ranum in InfoSec Magazine, May 2008

P.S. Thanks for lending us a whetstone, Marcus. =)

Table of Contents

:: About the authors 3
:: Introduction 4

:: Vivisecting The Social Net Model 5
:: Connections, Entities 5
:: Personae and Simulacra 5
:: Culture of Trust 6
:: Framework and Platform 6

:: Attacking Social Networks 7
:: Attacking Social Network Functionality 7
:: Attacking Via Offsite Content 8
:: Attacking with the IMG Tag 8
:: Request Forgeries and Social Nets 8
:: Attacking “Innocuous” Functions 8
:: SocEng and technical attacks 9
:: Attacking Social Network Applications 9
:: Attacking Via Social Network Applications 10

:: An impersonation experiment 11
:: Connections are currency, but currency is cheap 13
:: Good bait yields results 14

:: About the authors

:: Shawn Moyer is the founder of Agura Digital Security, a web and network security consultancy. He has led security projects for major multinational corporations and the federal government, written for Information Security magazine, and spoken previously at BH and other conferences.

Shawn is currently working on a slash fanfic adaptation of 2001:A Space Odyssey, told from the perspective of Hal9000. He only accepts friend requests on Facebook if they include a DNA sample and a scanned copy of a valid driver's license or passport.

:: Nathan Hamiel is a Senior Consultant for Idea Information Security and the founder of the Hexagon Security Group. He is also an Associate Professor at the University of Advancing Technology. Nathan has previously presented at numerous other conferences including DefCon, Shmoocon, Toorcon, and HOPE.

Nathan spent much of DefCon 15 without shoes and is planning ahead this year with a defense-in-depth approach that includes failover footwear. He has 1,936 people in his extended network, and finds that disturbing on a number of levels.

:: Introduction

Like most folks of a security bent (and if you're reading this, that probably means you), we've spent a lot of time watching Web 2.0 with bemusement.

Promiscuous sharing of information, client-side Javascript goop, blogging, mini-blogging, micro-blogging, vlogging, social nets and social media have all given the web much of what the starry-eyed latte-chugging idealists of Web 1.0 and the dot-bomb boom were yammering on about ten years ago: a platform for anyone to create content, to connect, to share, and to carve out a little space for themselves and a few million of their closest friends.

All of the above, of course, seems to run absolutely orthogonal to everything those of us in InfoSec preach: "Validate all user input. Authenticate and tokenize everything. Sanitize all output. Audit the crap out of anything before it goes live. Limit functionality to core functional requirements. Trust no one."

From a securability perspective, giving every user of an application a more or less open platform to create content and write their own (carefully sanitized and oh-most-assuredly vetted) apps, stylesheets and scripts to share with their Interweb penpals sounds like the lunatics running the asylum, doesn't it? Nobody can build a sandbox that big -- or rather an infinite series of sandboxes, with a series of little tunnels between them... Does your head hurt yet?

And yet, here we are. This year BlackHat and sister con DefCon invited attendees and speakers to join LinkedIn groups and are micro-blogging on Twitter (as are both of your intrepid authors). These are organizations whose members are often only known by their handles, who have lived through presenters being sued, arrested, and detained in airports, part of what is arguably the largest hacker community in existence, made up of some of the most paranoid netizens on the planet. Asking us to join their friends list? What gives?

Well, let's face it. This stuff is like crack. The ability to connect and communicate in a simple interface brings the human back into the digital. The thrill of someone accepting a friend request or responding to a message in many ways evokes the old feeling you got in the heydays of BBSing when you stumbled on a new board and saw someone you knew... Someone on the other side of the screen, out there, is listening to you. Someone else out there thinks you matter.

So if even those of us in the paranoid-by-profession security world are getting sucked in, and we accept this stuff isn't going away any time soon, how bad is it really? How far behind the usual safety versus features curve are we at this point? That's what we've been trying to work out.

:: Vivisecting The Social Net Model

To build a framework for how social nets can be attacked, we spent some time trying understand all of the moving parts. There are literally hundreds of social networking sites, but most share some basic functionality, with varying degrees of flair and features, from the very sparse (Twitter, Pownce, and social components of blogging sites and forums) to the staggeringly complex features arms race between Facebook and MySpace.

:: Connections, Entities

An entity is simply something that can share connections with other entities. It might be a bot, it might be a group or affiliation, it might be an app, or it might be a human. If it has a friends list, it's an entity.

The key here is that a connection between two entities implies some degree of trust, and each entity is a spoke that creates nth-degree connections. If you install an application, join a group, or add a connection, you've trusted that entity, and by association those in your connection list have some level of transitive trust of the entity as well.

Connections and links between entities are the meat and bones of social networks, naturally. Much of SocNet attack vectors come from here, and for us, the framework of connections mean that by definition, any social network contains a social component.

For our purposes, and with moves to interoperation and applications built on building connections between SocNets, we also consider connections to span not only a given site, but potentially to span other SocNets as well.

:: Personae and Simulacra

SocNets have a voyeuristic quality to them that seems to draw people in. Our SocNet profiles are about showing the world our perception of ourselves. Users make

liberal use of soft-focus pics and flattering angles -- the“LiveJournal head tilt” and the “MySpace angle shot” and create a thinner, richer, more clever simulacrum of themselves. On the business networking side, bombastic resumes and reciprocal endorsements from coworkers help build the identity of a model SocNet citizen.

Effective approaches of attack take into account this seemingly universal need to build up a better-than-IRL persona. Approaches that play on vanity, or provide a way to further build the persona will yield better results.

:: Culture of Trust

Building a platform based on user-created content and applications means that from inception there is a basic trust of the user population. Interestingly, though, unlike other social media like Wikis, there's very little democracy involved, and users have few recourses to police bad actors other than to send complaints to an always-full complaint mailbox.

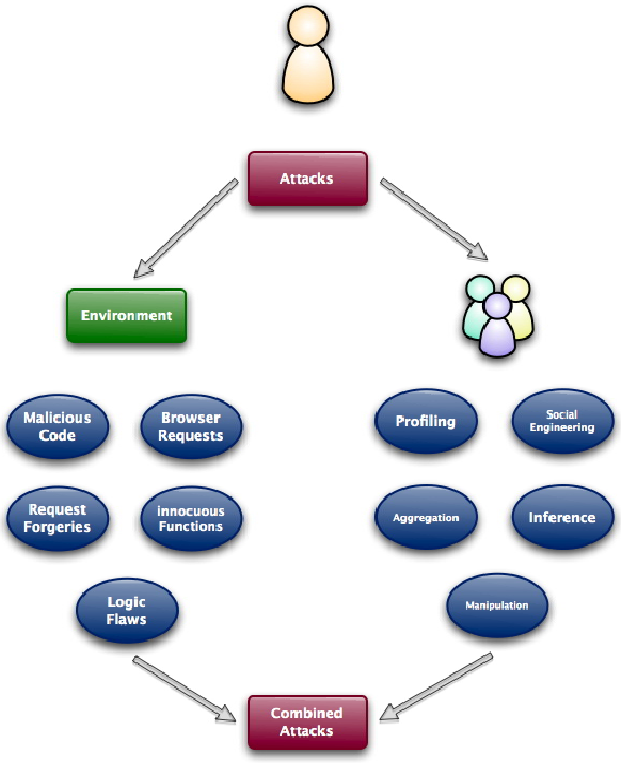
Since becoming a member of a SocNet requires the initial leap of creating at least a semi-public profile, it becomes a small step to move a bit further, and accept message and connection requests from people you haven't really verified, and to share further and further information. This pervasive culture of trust and sharing is much of what makes SocNets so appealing from an attacker's standpoint.

:: Framework and Platform

SocNet frameworks are built on facilitating connections, and then creating rapport between those connections. Providers are constantly extending this functionality and adding new features to match up potential connections, send messages and share media, creating more affinities and eyeballs for hyperfocused targeted advertising (which ultimately, is the reason Social Nets exist in the first place).

OpenSocial and other published application APIs have now made it far more possible to integrate social apps and share data between sites, and we're now seeing a move to cross-site functionality, and naturally, the creation of new shared exposure.

:: Attacking Social Networks



Social networks make great targets due to the fact they are large collections of individuals. Large collections of individuals usually equal large collection of potential victims. Social networks also provide an environment where victims are more likely to come to an attacker. Social network culture is a trusting culture. Typically people trust by default. Whenever you have an environment where individuals trust by default there is a high degree of attacks being successful. Individuals are less likely to think that attackers would target them or that anything they have would be worth an attacker's time.

Individuals in social nets may also have the perception that browser based software is less dangerous than installed software. Users are far more likely, warnings and disclaimers aside, to install social applications than they are to download and install a traditional executable.

As of late there has been a shift of focus for social networks toward business and professionals. More and more social networks are cropping up that have this focus. There are many professional networks that cater to business professionals, like LinkedIn, Plaxo, and Naymz, and Facebook has also worked at attracting some level of professional networking. This means that Social Networks can now also become a path into a corporate user population and a new vector for targeted attacks.

:: Attacking Social Network Functionality

A quick inventory of what an attacker has at their disposal when attacking social networks reveals quite a few possibilities. Social networks are constructed with a great majority of their content provided by their users. This content model can introduce many potential vulnerabilities. The more restrictions that are put in place, the less "open" the social network becomes. If the social network becomes too restrictive they risk losing their users to other networks.

With large numbers of users as members of social networks and an increasing membership these networks are going to continue to be attacked. Large numbers of users also make the impact of vulnerabilities much worse and raises the success of the for attacks.

:: Attacking Via Offsite Content

Offsite content cannot be controlled in any way by the social network. It is possible to hijack calls for this offsite content and steal the request. Once a malicious user steals the request it is at the mercy of the attacker where they want the request to go. It appears that many sites that do allow linking to offsite content allow it everywhere on their site allowing an attacker multiple opportunities to link to their content.

:: Attacking with the IMG Tag

The IMG tag is often used in XSS, via the ONERROR, ONCLICK, tags, etc, but even if a site sanitizes properly for XSS, links to offsite content could still be used to attack both offsite systems and the social network itself. With the IMG tag the browser basically gives you a free GET request. Even if file extensions are sanitized to .jpg, .gif, etc you can still use URL injection on another site or any number of free URL forwarding services to redirect the GET request anywhere, including back to the social networking site from which the request originated.

Since a high-traffic profile might get thousands of views a day, and since simply viewing a profile will cause the client to make a request, there are thousands of possibilities for using the IMG tag and other links to external content to forge request and attack clients. Essentially, with a bit of social engineering, any SocNet site that allows unverified links to external content, IMG tag or otherwise, place some level of control of the user into the hands of the attacker.

:: Request Forgeries and Social Nets

We identified several issues with regard to request forgery on a number of sites that we reviewed. These request forgeries could be either from the same site (SSRF???) or across sites as typical CSRF -- you view my page on one site, and our CSRF caused you to install an app, send a message, or add us as a friend on another site, etc.

:: Attacking “Innocuous” Functions

Certain functions are considered blatantly important and require protection. Developers often realize that account changes, profile changes, sending messages,

etc. require some form of protection, but do not consider other functions as important. There are other functions that seem innocuous that can be used to create some pretty nasty conditions.

Many built-in functions are not something that is typically seen as a “privileged” function and as such not seen as requiring safeguards. Logout functionality, for example, can be used to create a sort of denial of service against a target. This attack is done by creating a request forgery to the logout functionality of the site. When a user views content linked to this logout they are then logged out of the social networking site. This is an effective way to end someone’s SocNet profile, which could be useful in assuming their persona for oneself.

In the case of many social networking sites the content is often rendered to both the viewers of profile content and also the owners of profile content. This means that the administration to remove items such as malicious comments can be difficult because they are being logged out of that portion of the site prior to being able to delete the content.

:: SocEng and technical attacks

Both social and technical attacks can be combined to increase effectiveness of attacks. For example, if there were an instance where an attacker wanted to assume someone’s persona, they could launch a technical attack against the persona’s profile and follow it up by standing up a new profile. They would then attempt to re-add the individuals from the persona’s friends list and just state that something had happened to their profile.

A technical attack that would compliment this social attack would be an attack that blocks communication from the individuals account. With the individual no longer able to contact people on his friends list it might take some time to get noticed.

:: Attacking Social Network Applications

Social network applications can also be attacked in the same way another other web applications are attacked. The same vulnerabilities that plague traditional web applications are also relevant here. These applications are often created by individuals who are merely members of the social network and have very little programming or security background.

The prerequisites for creating and deploying applications on a social network differ between networks but can be as simple as just merely having five friends or just asking for the access. With such a low bar there is bound to be plenty of people with relatively little programming and security experience. This can put the information collected by the application at risk to compromise.

Application developers may unknowingly introduce vulnerabilities in to their

applications that would allow a malicious user to exploit them. Previously there have been vulnerabilities in social network applications that allow malicious users to take actions in the name of someone else. Although on the surface this may seem like nothing more than mischief, a compromise in the application's data could allow for an attacker to access private information of the user.

:: Attacking Via Social Network Applications

Social network applications add another layer on top of the installed base infrastructure. This allows developers to extend the functionality of the social network and add features that are not natively available to their users. There are many of these applications available and their functionality ranges from just displaying static content to actively taking actions with other users.

These applications make great delivery methods for attacking users for several reasons. The applications are rendered in a browser window and not installed on a user's computer. This rendering gives unknowing users a feeling of safety because they are not "installing" applications on their computer. Many users consider viruses and malware to be delivered when they install something on their computers.

It is easy to target people and get a high rate of installs on your application if you choose the right delivery method. People want to install things that are cool and popular. Frameworks that can be used across multiple social networks provide an attacker with an environment that is build once - literally, "write once, own anywhere". An attacker only has to build one malicious application and they can multiply their potential targets by installing them on supported social networks.

Social network applications have a an implied endorsement, of being published by the social network. Even though there are many statements to the contrary, to an average user it appears they get these applications from the social network itself. On top of this, the application's EULA absolves the social net from all responsibility.

:: An impersonation experiment

As we started thinking about some of the SocEng scenarios we wanted to run through for this project, something we kept coming back to was impersonation. The lack of validation and culture of trust in social networks seemed to be just begging for an impersonation scenario.

Social net personae have zero or less authentication -- zero in that you're taken on your word that you are who you say you are (and via the social proof of your persona's connections, which we'll get to in a bit), and less than zero in that it's trivial to change that persona or create a new one and become someone else.

A number of highly public figures, most notably presidential candidate Barack Obama, have had profiles created on their behalf without their permission. Obviously, the ongoing Megan Meier case has shown the potential destructive power a well-crafted and plausible identity on a Social Net can wield.

While in Obama's case the intention was benign, having access to so many clients (targets) is a powerful exploitative tool, especially if paired with a mechanism to execute code on the client like a trojaned app or even something as trivial as a link to site full of malware. To us, impersonating a high-profile person was especially interesting because the target has the same powerful pull that "Josh Evans" had for Megan Meier -- we all want to connect with someone out of our league, someone a few notches above us in the SocNet hierarchy. Celebrities, however minor, have a universal appeal, and connections to them add powerful credibility to the online personae that SocNet users strive to create for themselves.

We decided to engage in a social experiment to see how many connections we could build with a doppelganger, and how long it might take until the impersonation was exposed. As Dan Kaminsky has said, why weaponize the obvious? We decided that obtaining a sigificant number of credible connections was enough to prove the exposure -- once the connection and trust were built the number of exploitation vectors at an attacker's disposal were too numerous to list.

With the above in mind, we spent some time searching for public figures in the InfoSec community that weren't active on social networks. The resonance? We're supposed to be the paranoid ones, right? The voices of reason, the chasers of the foxes from the henhouse. Once we'd identified a shortlist, Shawn sent the following mail to each of our candidates, CC-ing BlackHat founder Jeff Moss to add some credibility to our excercise:

> *Subject: Becoming $INFOSEC_LUMINARY: (We would like to impersonate you!)*
>
> *I wanted to run this by you before we started to do some work. We met briefly following $RANDOM_ANECDOTE_FROM_SECURITY_CONFERENCE.*
>
> *Nathan Hamiel and I are security consultants and frequent security conference speakers, and are currently working on a talk about exposures in Social Networking. The talk has been submitted to BlackHat, as well as to DefCon.*

As someone with tremendous visibility and known views about security and privacy, we thought it would be especially interesting to impersonate, well... You. :)

Our intent is to point out that in addition to a number of technical issues we've found, Social Net sites have zero validation of real identity, and that as long as a plausible effort is made (bio, "candid" photo, etc), there's a staggering amount of trust open to exploitation in these environments.

What we're proposing is to create an alias online posing as you on a number of Social Networking sites, and see how many folks will accept a connection request, and how many will request a connection. Once the experiment is complete we will either delete the profile, or (if you prefer), hand the credentials over to you.

Our request to you, if it's amenable: Don't respond to any mails or phone calls asking if the profile is you or not. Simply file these away for a few months while we engage in the experiment.

Your thoughts?

A mail roughly in this format was sent to a number of security luminaries, and responses were a pretty mixed bag. Interestingly, Marcus Ranum (who totally rocks, by the way), was absolutely on board.

Interesting idea!!! And I'd be game / I am game if you want to. BUT there's a small hitch -- I've gotten dozens of those linkdn-style requests in the past, and have sent people a fairly memorable "NO THANKS" note. If you start trying to impersonate me, you may get back some "what did you change your mind?" messages. :)

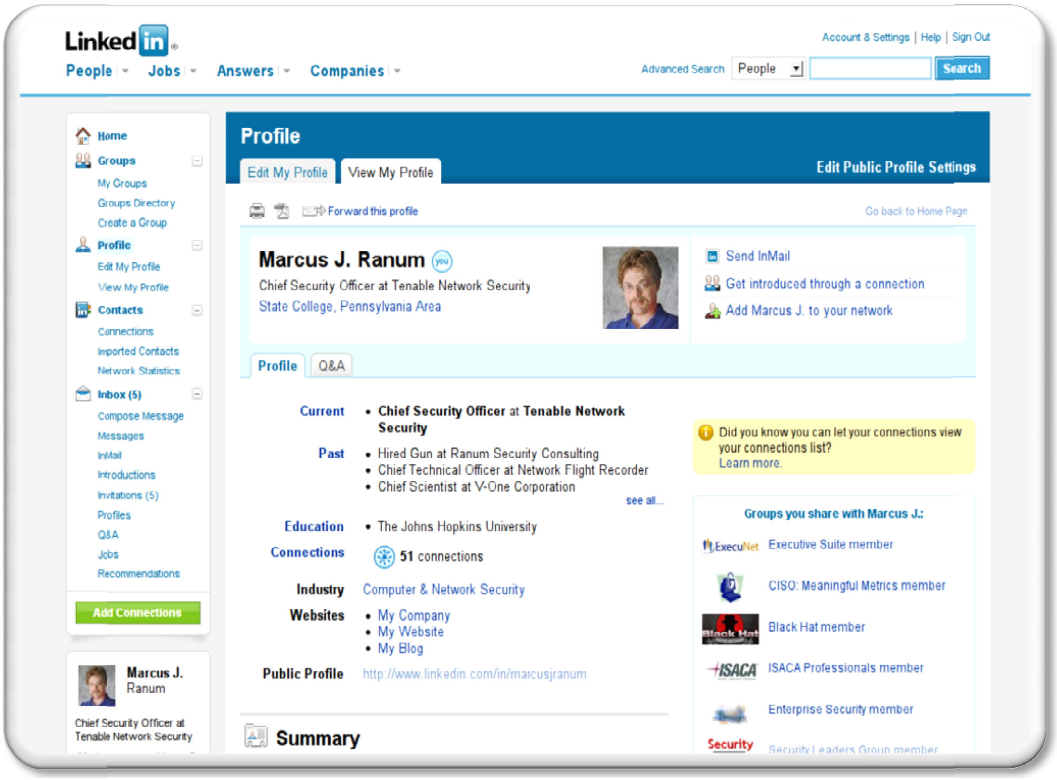
By the way, I agree about the crappy authentication. I have a number of alternative identities that I maintain, and they were ridiculously easy to set up. One site where I have an alt-ID actually decided to improve their authentication - but then grandfathered in all the current account holders. D'oh!

What works? Bootstrapping off a credit card or paypal? Eesh.

mjr.

We assured Marcus it was our job to perform the SocEng, and if his friends and accquintances didn't buy the gambit because we didn't "play" him properly, that was useful to the experiment as well.

A few weeks later we got started, and created our "Evil Marcus" LinkedIn profile:



Building a plausible profile and resume was trivial: we copied a PR photo from a recent conference engagement, bio information from Marcus' own website, and built a resume from Wikipedia entries and the inevitable press releases that were issued when Marcus joined companies over the past ten years or so. Some information was conjecture, but most was from publically available information.

:: Connections are currency, but currency is cheap

As soon as our profile was created, we started fabricating credibility by building enough connections to give the profile some credibility. Like most Social Nets, LinkedIn has a sort of parasitic underbelly of users that gleefully accept friend requests from anyone, on the somewhat bizarre pretense of "building a network" (apparently composed of other people doing the same thing), presumably for spamming links to vanity blog posts and headhunting -- most are either IT recruiters, or the usual Web2.0 suspects: bloggers, "career coaches", and "Internet entrepenuers".

On LinkedIn these people call euphemistically themselves "open networkers",

and are members of a number of groups, the most visible of which are TopLinked.com and LION ("LinkedIn Open Networkers").

A quick Google search gave us enough "open networkers" to build our connection list:

> *"invites accepted" OR "open networker" OR "accepts all invites" OR lion OR toplinked.com OR mylink500 +site:linkedin.com +inurl:/in/ -inurl:updates*

Within in a hour of creating our profile, we sent 50 or so connection requests. In 12 hours, we had 42 connections, a plausible enough number to make our profile credible. We then began joining security networking communities to build some more creditibility. By the middle of first day, our profile was a member of the *CISO: Meaningful Metrics, ISACA, Executive Suite, Enterprise Security, Security Leaders,* and *BlackHat* (which in Jeff's defense accepts requests from anyone -- we *were* asked for further validation to join the smaller *BlackHat Speakers* group) LinkedIn networking groups.

:: Good bait yields results

So our doppelganger was alive now -- a plausible, credible Marcus Ranum persona on a Social Network, in less than 24 hours. Our next step was to spend some time waiting for connection requests and bide our time before engaging in more directed social engineering.

Our first connection request came from the CSO of a security firm, within six hours of creating the profile. The next was the former CSO of a fortune 100 multinational. After that, we received a connection request from a member of Marcus' immediate family. Following that we made connections with a security consultant in Chicago, and the chief technical editor of a well-known security publication.

Within 24 hours, we had created a plausible profile and obtained rapport with high-value targets, with minimal effort. Not one person questioned the Marcus profile, and many users sent personal messages saying how excited that they were to see Marcus online. It seems obvious that if our doppelganger had sent a link to a new website, or asked the user to try out a new app, the success ratio would be substantially higher than a typical anonymous phish.