Studying offensive computing is essential

MAJ T.J. O'Connor, CPT Ryan Hand, CW3 Matt McDougall

The best defense begins with a strong offense.

An effective defense for successfully repelling threats to our networks must include a knowledgeable offense.

Combating the threats to our military systems is one of the most critical roles of the Signal Regiment. In the last six years, the number of reported cyber attacks has grown by 650 percent. Our adversaries' capabilities are exponentially multiplying attacks on networks.

As we Army communicators charge forward into this challenging domain of warfare, we must ask relevant questions. One of the hard questions to be asked and answered is "Do our Signaleers have an adequate basic understanding of the elementary tactics, techniques, and procedures in this domain of warfare?

For our officer professional development program, we began exploring some of the concepts involved in offensive computing, because we really don't know what we don't know about our adversaries' tactics.

Over the last 12 months, the Signal officers in our unit began taking the same classes as attackers, studying how our adversaries operate, analyzing the operational successes of organizations like Anonymous and learning to hack.

Studying concepts like penetration testing, exploit development, wireless exploitation and forensic recovery, we learned to attack exactly like the adversary. We traveled to compete and win hacker competitions, remotely attacked toy unmanned aerial vehicles during officer professional development lunches, and wrote and developed open-source attack tools. Twelve months of this professional development has brought us to some interesting understandings that we would like to share, in the hope that as a Regiment we can learn to defend this domain of warfare better.

Tried-and-True Means Tried and Exploited

All too often military experiences teach us to only apply tried and tested concepts in warfare. Consider when the Marines sought a viable rotor-wing aircraft to rapidly insert small teams into Afghanistan. Marine Detachment 367 selected the UH-1Y Huey, an updated version of a Vietnam-proven close-air support helicopter. With minor modifications to the weapons systems, the Marines built an aircraft capable of close-air support, small-team insertion and extraction, and casualty evacuation. Building upon what had already been proven during Vietnam; the Marines re-launched an almost retired air platform in less than two years. Consider this against the lengthy and tragic process of making the Marine Osprey a viable air platform, and you can begin to understand why military planners think this way.

However, these concepts do not translate to cyber. Let's examine why. After brief study as attackers we realized that defending Windows XP is nearly impossible. There is simply no way to patch a decade-old operating system successfully. Incremental versions of Microsoft's flagship operating system have included security mechanisms such as a randomized address space, prevention from overwriting exception handling, and a non-executable stack. These seem

(Continued on page 16)

(Continued from page 15)

like foreign concepts to a defender-only versed individual. However, to an attacker it means the degree of difficulty in writing an exploit program goes from requiring one high-school computer science class to a PhD in computer science.

Failure to understand this concept is not only a military-oriented problem. In April 2011, RSA confirmed that they had been compromised by a novel exploit for Microsoft Excel. The exploit infected several systems in the company, ultimately leading to the theft of the source code for their proprietary SecureID product. Later that summer, the same attackers used the proprietary code to attack RSA's customers, Lockheed-Martin and Northrop Grumman. A November 2011 research report concluded that the exploit would have failed if RSA had used a more modern version of the operating system that enabled hardware DEP by default. It is difficult to fully comprehend the concept that we should be using bleeding-edge operating systems instead of stable proven systems. It is a concept you can only truly understand when you learn to write your first exploit, which leads us to a second point: you must learn to write an exploit program before you ever learn to defend.

No Basis for Defense Without Attack

All too often in cyberspace we try to separate the concepts of attack and defense. Because of military authorities, clearances, and capabilities, we have separated the roles of each. For the most part, the Signal Regiment has taken the role of the Army's network enterprise defenders; but how can you truly consider yourself a defender without ever having attacked a system?

To understand this concept, consider the role of a young infantryman. He does not consider himself an offensive or defensive infantryman. He understands that the battle lines will shift over time and he is not fixed on one specific role. He may spend one day on the offensive, pursuing the enemy deep into his territory. The following day, he may be asked to guard a resupply convoy from the same enemy. As an infantryman, he teaches, mentors and coaches his subordinates on tactical movement and weapons systems, studying how either side of the battle may employ them.

Yet Army leaders and planners are largely drawing battle lines in cyberspace. This can only help to create the weakest defenders possible. How does an enterprise defender know how to look for indicators of a compromise if that person has never compromised a system? In 2006, the Pentagon disclosed hostile cyber units attacked our NIPRNET and downloaded up to 20 TB of data. The attackers used a technique of passing the hash from internal systems to grant unified access to co-located and co-managed systems. Only once finding sensitive data, the attackers compressed that data into compressed archives and pushed it outside our network to foreign file transfer protocol servers.

Twenty TBs leaving our networks is a needle in a haystack for an ignorant defender to classify as malicious, but to a trained attacker that needle is as obvious as the Empire State Building. Compressing data and pushing it to FTP servers to reduce a signature is a junior varsity attacker move. The fact that the same data went to foreign FTP servers would have been spotted by anyone who has ever attacked a system. Unrolling those clues and tying multiple remote process execution commands to them would confirm the attack. Like the infantryman who pauses to examine and disable a trip wire before assaulting an enemy's base, we must understand that our role in cyberspace is not clearly offensive or defensive. When we understand how to attack, we begin to see our defense surface much more clearly.

We Are Only Defending the Visible Attack Surface

Consider the defenses emplaced in your organization for cyber defense. How much did you spend ensuring that cabling was shielded from electro-magnetic emissions? Anyone who has ever built an Army network knows the immense struggle to accredit a facility to process SIPRNET traffic. We emplace a protective distributed system to deter and/or make difficult physical access to the communication lines carrying national security information. We ensure there are approved electronic locks on our network closets. We only procure equipment through reputable U.S.-only based vendors. Annually, organizations spend millions on these defenses. Why? These defenses are critical to our overall defense posture. However, we have a habit of only placing these defenses where we can physically see them. Looking at a locked comms closet with a biometric authentication, we feel like we are making adequate and complete defense. Defending only the physical visible attack surface can ultimately lead to failure. But we'd argue it is a mindset that is prevalent in today's Army. All too

often enterprise defenders think systems are safe if they are physically secured and patched with current updates and anti-virus programs. This is untrue.

Early in the Spring of 2012 we participated in a hacker tournament where we had to gain access to several unauthorized systems in a virtual environment. Lacking physical access, we had to gain system level privileges to a fully patched computer on a virtual enterprise network. Sounds difficult, right? No physical access, system fully patched, anti-virus program running. Should be good, right? No. Within minutes, we found a separate client workstation and sent the user a spam e-mail with a link to an unsigned malicious java applet.

The user clicked the link and ran the applet, granting us full permissions to that machine. At this point, we noticed the machine had an enabled local administrator account. Win! These happen to be the same administrator credentials necessary to log on to our ultimate target. Within minutes, we gained access to a fully patched, well-defended machine.

The attack space is clearly visible to an attacker. They attack things like unpatched remote services, client-side application vulnerabilities, weak credentials, and expose trust relationships. Most network defenders are pretty good about patching systems. However, a defender-only-versed individual fails to see the full defense space, such as ensuring they disable local administrator credentials. An attacker knows this, though, because the password and account policies are one of the first things he or she will examine after initially comprising a target. Let's examine another scenario where a weak password can trump even the best theoretical defense.

Implementation Trumps Theory

Most Army Signaleers are familiar with the Federal Information Process Standards Publications. Specifically, FIPS 140-2 contains some guidelines on purchasing IT products that contain cryptographic modules. For example, when purchasing a wireless access point that contains encryption, you must ensure that it complies with FIPS 140-2. Knowing that a body such as NIST has validated the cryptographic algorithms on a device can give an enterprise defender some level of comfort. However, again, it only serves as a false level of comfort for someone who does not understand the attack surface.

Recently, we asked a colleague to set up a secure wireless access point. After examining all his available options, he chose some sound security-related settings on the access point. He placed the wireless access point in hidden mode (ensuring the access point did not broadcast its network name), enabled MAC restriction (ensuring only specific MAC addresses could connect to the access point), and finally chose the WPA2 handshake-authentication with AES (ensuring that the traffic was prevented from eavesdropping or replay attacks). Outstanding! Our colleague configured the access point in a secure manner as best he understood it.

(Continued on page 18)

(Continued from page 17)

Next, our team attempted to gain access to the access point. We began by sniffing wireless traffic and saw the unencrypted management frames between the access point and our colleague's computer. In an option field of the traffic, we saw the hidden name of the access point. Next, we changed the MAC address of our machine to that of our colleague's computer. With the address changed, we forged a deauthentication packet, severing the original connection. We then watched and captured the WPA2 handshake as our client attempted to reconnect. Running the WPA2 handshake through a brute-force cracker, we noticed the colleague had used a dictionary word for the password. Our colleague was stunned to realize that the password played any role in the overall security of the access point.

While we praised our colleague for knowing all the available options for security, we equally chastised him for failing to choose a secure password. He immediately changed the password to a complex password. Noticing that our colleague had left the default network name as Linksys, we then attacked the complex password using password rainbow tables. Again, our colleague was surprised to realize that the network name played any part in the exchange of the symmetric key for the network. Not surprisingly, he had never attacked a wireless access point before.

There is No Silver Bullet for Defense

So, moving forward, you may ask: What tool should I be using? What can I do to defend my systems? Arguably, we have very good tools for locking down enterprise networks and emplacing host-based controls. However, a good attacker will find a way around them. A decent rootkit can hook the Windows API calls, essentially hiding itself from an antivirus program that scans the file system. Specially crafted fragmented packets can be used in a covert method to evade network capture and network-based intrusion detection systems. In the case of the recent Flame attack, digital signatures can be spoofed to impersonate legitimate software vendors. A layered defense is good—not placing all our eggs in one basket and using multiple network and host-based technical controls is a good strategy.

Continuous education is the most powerful tool we have. Yet, at every impasse, we have noticed individuals arguing for control – "this should be a 255S function," "only the 53 should be qualified as IAM Level 1," "a 25 series officer could never understand the complexity of a cyber attack; he should be a manager."

There is room for everyone in this domain of warfare. Single ownership of the problem will ultimately lead to failure. Right now, all of our Signal Soldiers need continuous and deliberate education in the domain of cyber war. It must be woven into every aspect of every professional course, training, and exercise. Cyber should closely mirror our Safety and Risk-Reduction programs. Similar to a young platoon leader filling out a risk-assessment card before conducting a rifle range, a young Signaleer should be forced to consider the cyber implications of standing up a new Web server for his unit. Only after repeated, deliberate efforts to learn more about this domain of warfare can we begin to start formalizing solid enterprise defenses against our adversaries.

MAJ T. J. O'Connor *is a 25A Signal Officer at the 10th Special Forces Group (Airborne). Prior to serving in his current assignment, TJ taught computer exploitation at the U.S. Military Academy and deployed to Afghanistan and the global war on terror with the 7th Special Forces Group.*

CPT Ryan Hand *is a FA 53 Systems Automations Officer at the 10th Special Forces Group (Airborne). Prior to serving in his current assignment, Ryan deployed to Iraq for fifteen months with the 63rd Signal Battalion.*

CW3 Matt McDougall *is a 255A Information Services Technician with the 10th Special Forces Group (Airborne). His previous assignments include the 75th Ranger Regiment and the Joint Communications Unit.*

Join the Discussion
https://signallink.army.mil