PowerShell

David Kennedy (ReL1K)
Josh Kelley (Winfang)

http://www.secmaniac.com
Twitter: dave_rel1k winfang98
About Josh

• Security Analyst with a Fortune 1000 --- Works with Dave

• Heavy experience in penetration testing, exploitation, web application security, vulnerability management, and incident response.

• Primary languages are Python…and now PowerShell 😊
About Dave

• Director of Regional Security for a Fortune 1000

• Heavy experience in penetration testing, exploitation, web application security, wireless and physical

• Creator of the Social-Engineer Toolkit, the Social-Engineer Framework.

• Heavy military background in Intelligence, deployed twice to Iraq and other middle east countries.
Special Thanks

Special thanks to IronGeek and Kathy Peters
Brief Intro to PowerShell

- Windows version of a bash shell in nix... Very powerful, flexible, and getting improved regularly.

- Installed by default on all Windows 7 and Server 2008 operating systems. Full integration for all new existing Microsoft products, including Exchange and AD integration.

- Full integration into the .NET framework and can be directly called when performing scripting.
If you haven’t seen it..
PowerShell for h4x0rs

• We will be the first ones to admit the usefulness and power of PowerShell in a positive manner. The ability to perform advanced tasks on Microsoft based operating systems is a huge leap forward.

• PowerShell for us as security researchers can be a great addition ranging from tool creation and automation when performing security assessments.
Execution Policies

• Restricted – Places it in a mode where only certain scripts and calls can be called.

• AllSigned – This script only allows signed scripts to be executed. Has to be from a trusted publisher. This is the most restrictive policy.

• RemoteSigned – Remote scripts must be signed by a trusted publisher, things run locally don’t need to be signed.

• Unrestricted – Can run anything both remote and local.
Release of Metasploit Module 1 – PowerShell Debug

- Traditionally post-exploitation phase, if you didn’t have direct access to memory, traditional methods of getting a payload onto a system was through Windows debug (now removed in all newer operating systems), vbscript, csc, TFTP, or FTP.

- These methods are now proving much more difficult with better A/V and HIPS detection (well kinda..) and TFTP and FTP blocked egress.
DEMO – Metasploit Module
Small Example of Conversion

- Binary is converted to hexadecimal and placed onto the filesystem.

- Convert script is created to take the hexadecimal and rewrite it back in a byte array as binary.

- Payload is now on the system for execution.
Execution Restriction Policies

• Shouldn’t be relied upon for protecting execution of PowerShell based commands.

• Execution restriction policies do not help from a post exploitation perspective…
CreateCMD Release

• Contents of a file are concatenated, compressed, and converted to base 64 into a single string.

• A boilerplate bootstrap code created for powershell – Command or –encodedCommand args then unpack the code and then perform an Invoke-Expression

• That will execute the script contents in the current shell context with all new functions that are in the script.
What’s this mean…

• With the most restrictive policy set on PowerShell we can still execute whatever we want (again not a security prevention method).

• No need to disable execution restriction policies

• No registry interaction, no reboots, nothing.
DEMO – CreateCMD
What we can do..

• Since we have full access to both PowerShell and the .NET libraries, we can do pretty much anything we want which is great...

• Releasing today both a bind and reverse shell programmed purely in PowerShell as PoC tools that you can create for security testing and to demonstrate the power of what we have...
DEMO – PowerShell based SAMDump
PowerShell SAMDump

- Meterpreter based module, will dump the SAM database purely through powershell.

- Works on all operating systems, both x86 and 64 bit.
SET v0.6 – Codename

“Arnold Palmer”
Basics of SET

- Open-Source purely Python driven.
- Integration into Metasploit for both the exploit repository for client-side attacks and payloads.
- Multiple attack vectors specifically designed for Social-Engineering.
- For good, not bad, help pentesters and organizations test their security program.
SET DEMO

USB HID Attack Vector
USB HID Attack Vector

- Drop a payload onto a system either through PowerShell or WSCRIPT.
- Automatic creation of attack vector through SET
Integrating into Existing Hardware

- Most new keyboards have integrated USB Hubs.
All put together…

- Keyboard still works perfectly… We have our malicious stuff just sitting there waiting…
SET DEMO

Java Applet Attack Vector
Thomas Werth Attack Vector

• Released at ShmooCon, this attack vector allows you to create a malicious Java Applet.

• User hits “run” and the payload is executed on the victims machine.

• Redirects user back to original site to make attack less conspicuous.

• New in SET v0.6, heavy obfuscation of java and payload for A/V bypass and fixed major issues with Linux/OSX payload deployment. Applet source just opened today!
What does this mean?

• Anti-Virus and HIPS aren’t picking up these types of attacks, which means it’s a safe passage for exploitation.

• The usefulness of this really aids us in post-exploitation scenarios and for security research and analysis.
Future Plans

• Process injection and code injection capabilities within PowerShell.

• Ability to deploy security baselines to multiple systems and ensure enforcement.
Coming Soon

Louisville, KY, USA

DerbyCon

September 30th - October 2nd, 2011
Hyatt Regency Louisville
Contact us, info@derbycon.com

© Derby Con, All Rights Reserved | Designs by DigiP
Social-Engineer.org

- [http://www.social-engineer.org](http://www.social-engineer.org)
Thanks to Kathy Peters

Be sure to check out:

http://www.secmaniac.com

Twitter: dave_rel1k winfang98