Vulnerabilities in RAM Core Dumps

Joshua Hulst

04/20/09

1. Problem Definition

As the computing industry evolves, practices which were taken for granted in the past have to be re-examined and updated to work with modern systems. One such area which needs to be evaluated is the use of unencrypted data stored in RAM. Recent attacks such as the RAM cold boot attack (Skorobogatov) have demonstrated that RAM is not as secure as once thought. Another way to access contents stored in RAM is to crash the application and obtain a snapshot of the processes memory, also known as a core dump. In this report, I show how to setup an Linux to give core dumps and explain how this could be exploited to provide sensitive data to a malicious user. I also examine two popular open-source applications to see what data is available in their memory snapshot.

The main motivation to keeping data in RAM is the performance increases it provides. By keeping a certificate or password in RAM, the data can be used easier, rather than going to disk, which can be encrypted. The operating system is responsible for protecting RAM from other processes, providing a secure working environment for each process. But, some operating systems can be manipulated to give up RAM contents on abnormal termination of a process. Linux provides this functionality for debugging purposes. By examining the core dump, a developer can see problems and try to avoid them.

Linux capabilities to provide a core dumps can be manipulated though. If an attacker were to have access to an account which was running a secure application, they could crash the program, grab the core data, and analyze it. For instance, Eve works for a company that sells products and accepts credit cards as payment. Eve supports the web application which accepts the credit cards so she has the login of the user which the application runs as. Eve could perform the following steps and possible gain credit card information.

1. Eve makes application start with options to create a core dump on termination (Using ulimit)
2. Eve waits until users begin entering credit card information, which is then stored in RAM
3. Eve sends a signal to the process (using kill -11)
4. Application crashes, creating core dump
5. Eve analyzes core dump using a hex editor, finds confidential credit card data

2. Solution Approach

RAM is not secure. By keeping this in mind, the application developer can work to avoid storing sensitive data. It seems that many cases could be solved by clearing out the RAM data when done with it. The RAM which contains the data can be cleared out as soon as it is no longer needed. The shorter the time it resides in RAM, the smaller the chance that it will be in RAM when the core is dumped. For example, the following psuedocode could be written to erase the Credit Card number

```
char creditCard;
creditCard = readCCNum();
doStuff(creditCard);
creditCard = 0;
```

By doing this, the credit card number is stored only long enough to be processed, then erased.

3. Proof of concept case studies

The basic steps to get a core dump in Linux are very generic:

- Turn on core dumps – Use `ulimit -c unlimited` or place the line in /etc/sysctl.conf

    - Using ulimit will set the core dump for the current session, adding the line in sysctl.conf enables it system wide
- Start application
- Make the application segfault
    - This can be done by exploiting the application or by sending a signal using `kill -11 <pid>`
- The core dump will be created in the directory where the application was started named core.<pid>

3.1. Asterisk

Asterisk is a popular open-source Voice over IP server. It supports multiple protocols, including SIP. The SIP protocol allows for user authentication, using a provided password. After a user authenticates, the password is kept in RAM stored close to the username. The version of Asterisk used is 1.4. See Fig. 1 for the setup and steps performed.

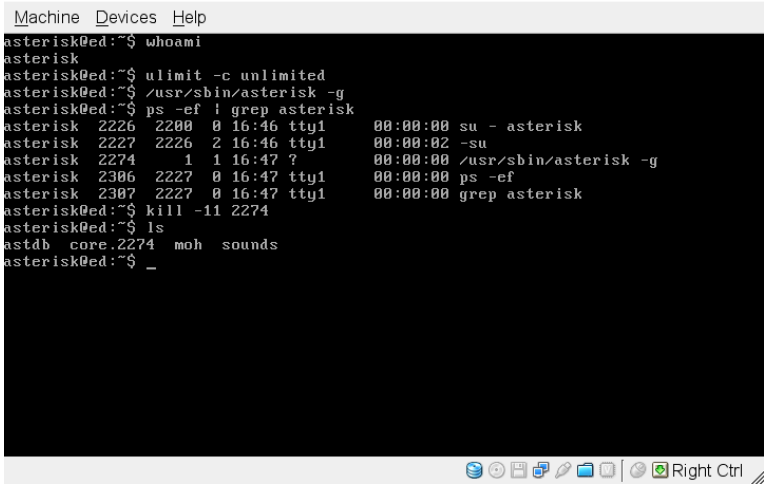
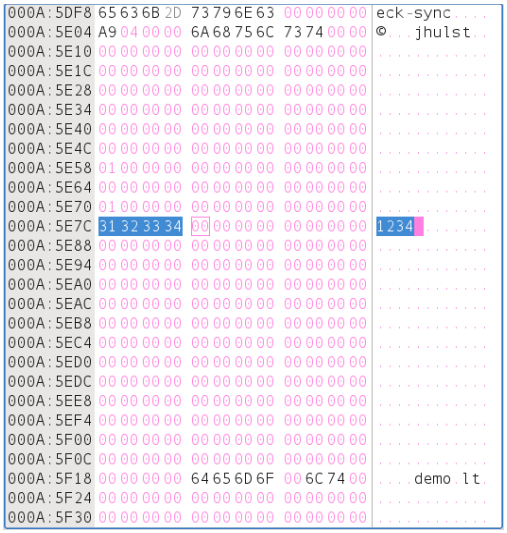


Fig. 1

Now that the core dump is obtained, it can be examined with a hex editor. In Fig. 2, Okteta is used. jhulst is the username, 1234 is the password used, and demo is the context of the user.



As can be seen, the user and all sensitive information is stored openly in RAM. Knowing this information would allow an attacker to impersonate a user.

3.2. Apache

Apache is an open-source web server. As with many other Linux programs, it gets its configuration from a text file. To get Apache to produce a core dump, a configuration option must be added to its configuration, which is commonly found in /etc/apache2/apache.conf. Adding the line CoreDumpDirectory <directory> instructs Apache to put the core dump in the specified directory. Core dumps obtained from an Apache instance running version 2.2.11 have plaintext versions of the configuration files. These configuration files include passwords to database servers as well as other services.

Figure 3 shows the option in the configuration file and Apache being started. It is then sent a kill signal using a method which sends the signal to all the child worker processes.

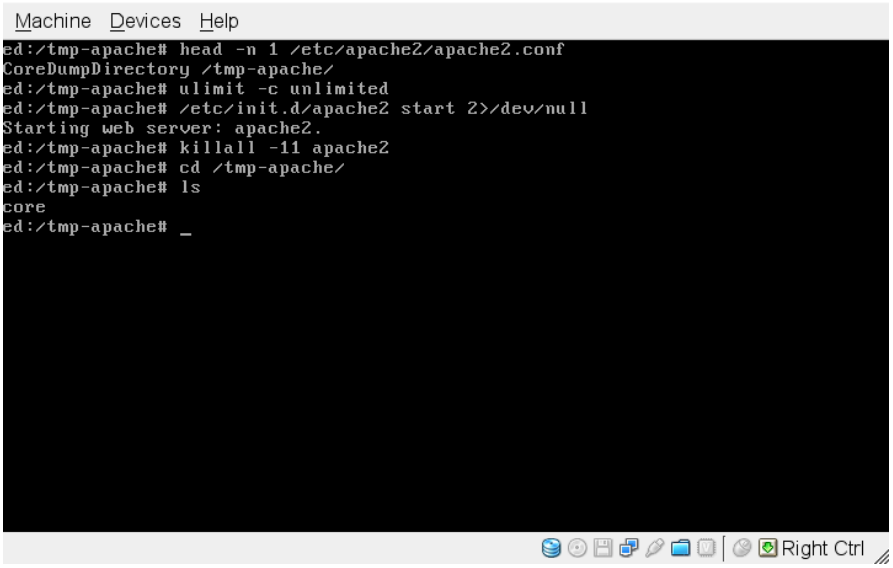


Fig. 3

As with the Asterisk example, the core dump can be loaded into a hex editor and analyzed. Here the mysql password and login (mythtv/mythtv) is seen as well as other parts of the configuration file which Apache uses to get all configuration options.

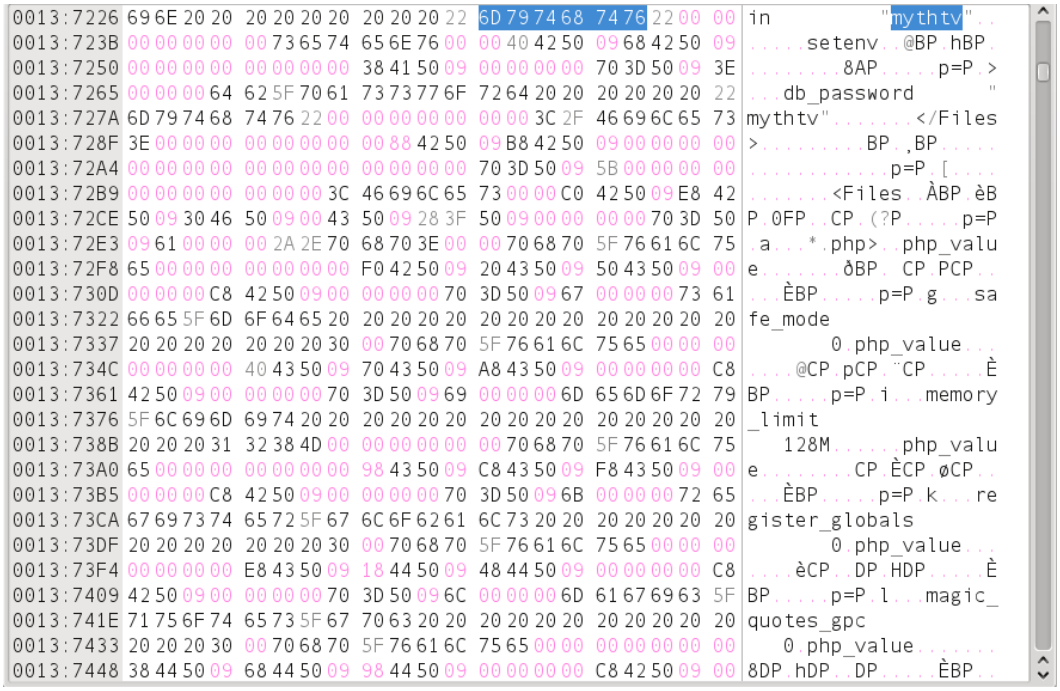


Fig. 4

4. Analysis

After obtaining the core dumps, a malicious user can analyze them using standard hex editors. By searching the data for known keywords, such as usernames, the user can locate areas where sensitive data is stored. In Asterisks case, as the password is stored close to the username, it is trivial to find the password once the username is known. Also, searching for large strings of printable characters is a good way to look for certificates or configuration files that are stored.

As the core dump retrieved is not necessarily time sensitive, the user has as much time as is necessary to perform full analysis. A suggested practice would be to obtain the core dump and transfer it to another machine for further analysis to be done at the users convenience.

While Linux provides a mechanism for obtaining the core dumps, there are restrictions imposed. First, only the user which owns the process is allowed to send the segmentation fault signal. Second, the core dump is created with read only permissions, owned by the user who owns the process. These restrictions are removed for the root user, but for everyone else, access is limited to user who owns the process. There are still cases where shared users can exploit the core dump, but its usefulness is limited.

5. Conclusion

As can be seen, the vulnerability of RAM is a problem which must be addressed. While the above methodology is limited in its usefulness, it exposes the problem further. RAM encryption is an option, but it must be unencrypted at some point. If this unencryption is done at the operating system level, the core dump method would still be valid. As with any vulnerability, there will be both hardware and software solutions to the problem, but until then, developers can be more careful about what is stored in RAM and how they use that data.

Works Cited

A Reference Guide to all things VoIP. 22 Apr. 2009 <http://www.voip-info.org>.

SIP: Session Initiation Protocol. Network Working Group. 23 Apr. 2009 <http://tools.ietf.org/html/rfc3261>.

Skorobogatov, Sergei. *Low temperature data remanence in static RAM*. University of Cambridge, Computer Laboratory. http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-536.html. Retrieved on 2008-02-27.