Project :: Defeating SkyNet
Part 2: Commanding the Legion

ELEC5616: Computer & Network Security
Luke Anderson *luke@lukeanderson.com.au*

May 4, 2018

Final due date is yet to be determined)

Introduction

In the first part, you created the base foundations of a botnet. Project 2 extends upon this and implements some core security features most recently used in the Conficker worm. Using these cryptography methods, Conficker was able to remain out of the control of hackers, well funded organisations and even the US government.

The new features for SkyNet include:

- Using public key cryptography to ensure only data or updates sent out by the botnet master are downloaded by the bot
- Using public key cryptography to ensure no-one but the botnet master can decrypt valuable data sent by the bot

SkyNet will be using advanced cryptograpy to protect itself against well funded organisations, government agencies and other hackers. What you will be implementing here is inspired by real world cases such as the Conficker work. To have any chance of defeating such a threat, you need to understand how they defend themselves.

Part 2 :: Protecting the Castle

1 Securely Updating SkyNet

In the previous part of the project, uploads were supplied to the bot via another bot using peer-to-peer (P2P) or a central website (pastebot.net). As you might have noticed, these updates were not verified in any secure way. By default, updates only needed to start with a specific string (Caesar) to be considered ‘signed’ from the botnet master. Any third parties with trivial reverse engineering knowledge would be able to realise this and create their own ‘signed’ updates.

You must devise a scheme where the botnet master is able to securely sign updates for SkyNet. Bots should perform this verification on updates retrieved via P2P or a website.

The exact mechanism by which this occurs is up to you, though some form of public key cryptography is suggested. Your signature scheme should also remain secure even when confrontd with attackers who have access to significant resources. The scheme should also be secure even if an attacker reverse engineers your program or the source code for your bot is stolen or released.

2 Securely Transferring Valuable Data to the Botnet Master

As part of their operation, the bots in SkyNet collect valuable data that is then sent to the botnet master. In the current codebase, this valuable data is uploaded in plaintext to pastebot.net. As it is plaintext, the data could be read by anyone who is able to intercept or access these uploads.

You must devise a scheme where the bots are able to securely upload valuable data to pastebot.net. No-one but the botnet master should be able to read the contents of these uploads. The exact mechanism by which this occurs is up to you, though some form of public key cryptography is suggested.

3 Implementation

An insecure skeleton framework written in Python 3 has been provided for you as a starting point. If you wish to use another language, such as Java with the Java Cryptography Extension (JCE), you may do so after seeking permission from your tutor. We can not provide any technical support if you select another language however.

4 Code Checklist

- Enable signing and verification for any botnet updates. This requires signing code in master_sign.py and verification code in the verify_file function in lib/files.py.
- Ensure upload_valuables_to_pastebot securely encrypts the data so it's only accessible to the botnet master. You also need to modify master_view.py to allow the file to be decrypted and read by the botnet master.

We will test that you can:

- Create and sign a new botnet update
- Test bot update verification by trying to download a legitimate update and a fradulent update from pastebot.net
- Test bot update verification when downloading updates via P2P
- Upload valuables to pastebot.net in an encrypted manner
- Decrypt encrypted valuables using the botnet master's private key

Your code must be well commented and in neat order.

5 Documentation

You are to write a two page design document outlining the security you implemented with your system. Your choices for authentication, confidentiality and integrity for the SkyNet botnet should be justified.

Specifically, you should provide a brief answer to these questions:

- How do you ensure the only one who can send updates to SkyNet is the botnet master?

- How do you protect the valuable information to ensure it can only be read by the botnet master? Remember that anyone can read the information uploaded onto pastebot.net.
- How do you ensure the botnet updates signed by the botnet master cannot be forged or modified?
- If SkyNet's botnet code is dismantled and/or the source code for it stolen, does your scheme become less secure?
- Give an indication of how difficult it would be for an adversary to take control of SkyNet when your protections are used.

6 Marking of Part 2

Once again, you will be submitting your code modifications for Part 2 to Phabricator. You do not need to create your own differential this time; instead your last commit to your repository before the due date will be used.

Similarly, as with Part 1 you should submit your report via canvas. A link will be provided closer to the due date.

7 Classmate Feedback for Part 1

Over the coming week, you will see that you will have been granted access to two additional differentials via Phabricator and identified as a "reviewer".

You must provide feedback for your allocated classmates on their code for Part 1 of the project. This is an individual task, not a group task. If you fail to provide meaningful and insightful feedback as requested, marks may be deducted from your individual mark.

Further instructions and guidelines around how you should provide feedback will be provided to you in a separate document in the near future.

Disclaimer

This is not an operational botnet nor do we intend you to create one. To defeat blackhat hackers, you must understand how they work and the tools they use. Recent botnets have used advanced computer science and cryptographic methods in order to remain secure from both hackers, well funded organisations and even governments. These advanced methods are what we intend you to learn and what we believe will help you detect, prevent and disassemble such attacks in the future.