Lecture 12
Packers and Cryptors

NMSU RE
Spring 2024
Joshua Reynolds

Malware Obfuscation

Used to:

- Hide C2 traffic
- Hide payloads from antivirus scanners

Weak Forms of Obfuscation - Base64

Base64 Encoding:

- An encoding invented to hold arbitrary binary data when sending over protocols or saving in databases that only support printable ASCII.
- Easily recognized because it will be a string of printable ASCII, often terminated with ‘=’ padding characters.
- B64 uses [A-Za-Z0-9+/], and is padded to a multiple of 3 bytes with “=”
- This is not (meaningful) encryption
- How much information per character does this encoding carry?

Weak Forms of Obfuscation - Custom Base64

Custom Base64 Encoding:

- Sometimes attackers try to be clever by swapping “+”, “/”, and “=” for other symbols.
- Other times, they may use URLBase64 which uses “_”, “-” and “=” for padding.
- Why would that encoding make more sense for URLs?

Weak Forms of Obfuscation - Caesar’s Cipher, ROT

Caesar’s Cipher:

Swapping bytes according to a list of rules like $ for A, G for F, @ for 4, etc.

ROT13

Shift each letter 13 spaces up in the alphabet, wrapping around the end a Z back to A.

ROT (General)

Like ROT13, but maybe using a different alphabet, or a different value than 13

Weak Forms of Obfuscation - Single-Byte XOR

You saw this in the Ransomware assignment.

Often, null bytes are skipped. Why?

How could you solve these, in general?

Crib:

When you know what the plaintext ought to decrypt to

Using an XOR Crib

$M_1 \text{ ^ } K = C_1$

Therefore:

$C_1 \text{ ^ } M_1 = K$

And we can use the key to do:

$C_2 \text{ ^ } K = M_2$

Weak Forms of Obfuscation - Multi-Byte XOR

This was in the graduate Ransomware assignment.

How could you solve these, in general?

Can you still use a crib?

Can you guess-and-check?

Can you use static analysis?

Can you use dynamic analysis?

Packers

Packers are tools that compress the size of a binary, and were originally helpful for things like getting your program to fit onto one CD for sale and distribution.

The fact that they obfuscate malware is a side effect.

Antivirus scanners probably unpack known packing tools, but won’t know how to do it for custom packing tools.

But, every packer must come with a stub program to unpack and run the payload, or it would not work.

Unpacking Custom Packers

Dealing with a custom packer requires some manual steps, with the help of the right tools for the job:

1. Using static analysis, find where the unpacker terminates and jumps to the **Original Entry Point**
2. Run the sample in a debugger, and set a breakpoint at that jump
3. After the unpacking stub runs, the full binary will be available in memory
4. Dump the process memory to a file.
5. Use a tool like Scylla to carve out the original executable that was packed.
6. You will need to help Scylla out with the address of the original entry point, and curate the list of functions that should be in the import table (IAT)

Cryptors

Cryptors are like packers, but they use real encryption algorithms, not just compression, to hide their contents.

They can be defeated in the same way as packers, because they must include enough code to undo the encryption at runtime.

They may source decryption keys from somewhere online to make analysis more difficult, which would require allowing it to talk to the internet (bad idea), or setting up FakeNet to capture the request, relay it yourself after checking if it contains any sensitive info, and pass the result on to the cryptor.

Detecting Cryptography

Crypto signatures – code segments that, when disassembled, perform operations that match the pattern of known cryptography libraries.

Crypto Keys – Generally look like X number of random-looking bytes in a variable or resource.

Intense Obfuscation

Highly obfuscated code may not be packed or encrypted, but just difficult to analyze due to:

- Building control flow with gotos
- Code that performs one function but when you jump into the middle of a multi-byte machine code instruction it does another
- Violating C calling conventions
- Using the stack in non-standard ways

How most obfuscation is applied

Using a tool someone else wrote, not writing a custom cryptor, packer, or obfuscation method.

There are tools that will do this for the attackers, such as…

MetaSploit Framework

Penetration testing tool with teeth

Contains thousands of known exploits

Contains obfuscation techniques to choose from

Can be combined to deliver other malware, or set up a callback shell

Requires little sophistication: can scan an IP address for vulnerable programs, exploit them, and inject the payload.

When a CVE is announced, it’s only a matter of time before MetaSploit can use it.

Script Kiddies, Hacker, and Exploit Researchers

- Script Kiddie:
  - Someone who uses tools written by others (like Metasploit) without understanding how they work
  - This term is derogatory, and implies immaturity as well as incompetence
- Hacker:
  - Someone who uses tools written by others, and understands them enough to adapt them when they don’t work off the shelf.
- Exploit Researcher:
  - Someone who builds new exploits, and writes the tools that others use