بسمه تعالی

# عنوان مستند

# بررسی بدافزار Quimitchin

فهرست مطالب

# ۱ مقدمه

باگذر زمان و رشد تکنولوژی، استفاده از فناوری اطلاعات به بخش انکارناشدنی از زندگی تبدیل شده است. با این وجود از همان ابتدای پیدایش، مواردی بودند که این عضو زندگی بشری را به مخاطره می‌انداختند. یکی از این مخاطرات بد افزارها می باشند. بـدافـــزار- مخفف عبـارت نرم افزار بـدخـــواه - (Malware) یک اصطلاح فراگیر و جامع است که به هر برنامه نرم افزاری اطلاق می‌شود که عمداً برای انجام اعمال غیرمجاز و گاهاً مضر ایجاد شده است. بد افزارها در قالب‌های مختلفی خود را نشان می‌دهند که تروجان‌ها و ویروس ها دسته های مشهوری از این خانواده هستند. در این گزارش می‌خواهیم به بررسی خانواده دیگری از بدافزارها که به یکی از چالش‌های بزرگ تیم‌های امنیتی تبدیل گشته، بپردازیم.

# ۲ بکدور چیست؟(Backdoor)

در دنیای امنیت بکدور یا همان درب پشتی به راه یا راه هایی گفته می‌شود که بتوان از طریق این آن به یک سیستم دسترسی غیرقانونی داشت. درهای پشتی را به سه دسته: فعال، غیرفعال و حمله‌بنیان تقسیم می‌کنند. درهای پشتی‌ای که منتظر رسیدن دستورات از طریق درگاه‌ها می‌شوند را غیرفعال می‌نامند. در مقابل درهای پشتی فعال خودشان آغازگر ارتباط با میزبان‌های دیگر هستند. درهای پشتی حمله‌بنیان به درهایی گفته می‌شود که با استفاده از حمله‌ای مبتنی بر یک بهره‌جو و با روش‌هایی مانند سرریز میان‌گیر به دسترسی‌های لازم در میزبان هدف دست می‌یابند.


شکل (۱)

# ۳ بررسی بدافزار Quimitchin

در بخش قبلی به بررسی تعاریف ابتدایی پرداختیم. در این بخش قصد داریم به بررسی اجمالی بدافزار Quimitchin که اولین بدافزار کشف شده برای سامانه‌های مک در سال ۲۰۱۷ است، بپردازیم. اولین سرنخ‌های کشف این بدافزار زمانی دیده شد که یکی از مدیران فناوری اطلاعات عبور ترافیک غیرعادی از سامانه‌های مک موجود در آن سازمان را مشاهده نمود. با بررسی‌های بیشتر انجام شده توسط محققان امنیتی این بدافزار کشف شد. این بد افزار به ظاهر خیلی ساده می‌باشد زیرا که از دو فایل plist.و .client تشکیل شده است که در شکل شماره۲ نمایش داده شده است.

```
~/.client
SHA256: ce07d208a2d89b4e0134f5282d9df580960d5c81412965a6d1a0786b27e7f044

~/Library/LaunchAgents/com.client.client.plist
SHA256: 83b712ec6b0b2d093d75c4553c66b95a3d1a1ca43e01c5e47aae49effce31ee3
```

**شکل (۲) : شکل کلی بدافزار**

فایل plist عامل راه انداز بدافزار می‌باشد و فایل client. را همیشه در حال اجرا نگه می‌دارد.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
 <key>KeepAlive</key>
 <true/>
 <key>Label</key>
 <string>com.client.client</string>
 <key>ProgramArguments</key>
 <array>
 <string>/Users/xxxx/.client</string>
 </array>
 <key>RunAtLoad</key>
 <true/>
 <key>NSUIElement</key>
 <string>1</string>
</dict>
</plist>
```

**شکل (۳): فایل plist**

علاوه بر فایل‌های مذکور، این بدافزار شامل فایل‌های دیگری است که به زبان پرل نوشته شده‌اند و از طریق همین اسکریپت‌ها با سرور کنترل خود ارتباط برقرار می‌کند. این اسکریپت همچنین شامل برخی از کدها برای گرفتن عکس از صفحه نمایش از طریق دستورات shell می‌باشد که به این منظور از دستور "screencapture" در مک و دستور "xwd" استفاده می‌نماید. همچنین با استفاده از دستور "uptime" در مک و یا دستور"cat /proc/uptime " لینوکس می‌تواند مدت زمان روشن ماندن سیستم را به دست بیاورد.

جالب ترین بخش از این اسکریپت، قسمت DATA است که در این قسمت سه فایل Mach-o ( فایل باینری مک)، یک اسکریپت پرل و یک کلاس جاوا با یکدیگر ترکیب می‌شوند. سپس این اسکریپت فایل ها را استخراج میکند و در درون پوشه temp ذخیره میگردد.

```
my $i = join('', < DATA > );
my %j = ('cr' => substr($i, 0, 72132), 'proxy' => substr($i, 72132, 1345), 'client' => substr($i, 73477, 3389));
my %k = ('cr' => '/tmp/cr', 'proxy' => '/tmp/proxy', 'client' => '/tmp/client.class');
my %l = ('cr' => '/tmp/', 'proxy' => 'cd /tmp && perl ', 'client' => 'java -Dapple.awt.UIElement=true -cp /tmp client');
```

**شکل (۴)**

از فایل باینری به منظور تصویربرداری از صفحه و دسترسی به وبکم رایانه استفاده می‌شود. که در این بدافزار از تعدادی تابع و فراخوانی‌های بسیار قدیمی که در شکل ۵ آمده، استفاده شده است .

```
SGGetChannelDeviceList
SGSetChannelDevice
SGSetChannelDeviceInput
SGInitialize
SGSetDataRef
SGNewChannel
QTNewGWorld
SGSetGWorld
SGSetChannelBounds
SGSetChannelUsage
SGSetDataProc
SGStartRecord
SGGetChannelSampleDescription
```

**شکل (۵)**

برخی از توابع به کار رفته بسیار قدیمی هستند و اکنون می‌توان گفت که تقریبا منسوخ شده‌اند. علاوه بر موارد مذکور، فایل باینری شامل سورس کد libjpeg که برای فشرده‌سازی تصاویر کاربرد دارد، است.

کلاس جاوا استفاده شده، به نظر می‌رسد قادر به دریافت دستورات برای انجام کارهای مختلف که شامل یکی دیگر از شیوه‌های عکس گرفتن از صفحه نمایش، گرفتن اندازه صفحه نمایش و اشاره‌گر موقعیت ماوس، تغییر موقعیت ماوس، شبیه‌سازی کلیک‌های ماوس و شبیه سازی پرس‌های کلیدی است. این کامپوننت به منظور فراهم کردن مکانیزم کنترل از راه دور (البته در سطوح ابتدایی و پایین) فراهم شده است.

```
case 16: // '\020'
    int i = translatekey(datainputstream.readByte());
    try
    {
        (new Robot()).keyPress(i);
    }
    catch(Exception exception2) { }
    break;
case 17: // '\021'
    int j = translatekey(datainputstream.readByte());
    try
    {
        (new Robot()).keyRelease(j);
    }
    catch(Exception exception3) { }
    break;
```

**شکل (۶)**

همچنین مشاهده شده است بدافزار حین اجرا در حال دانلود یک اسکریپت دیگر به نام "macsvc" از سرور کنترل کننده خود است. این اسکریپت از mDNS به منظور تهیه نقشه‌ای از تمام دستگاه‌های دیگر بر روی شبکه محلی و دادن اطلاعات در مورد تجهیزات موجود در شبکه از جمله IPv۶، IPv۴ و نام آن بر روی شبکه و پورتی که در حال استفاده از آن است، می‌باشد. علاوه بر جمع‌آوری این اطلاعات، این اسکریپت اقدام به برقراری ارتباط با دستگاه‌های پیدا شده، می‌کند.(شکل۷)

```
macsvc SHA256: b556c04c768d57af104716386fe4f23b01aa9d707cbc60385895e2b4fc08c9b0
```

**شکل (۷)**

همچنین علاوه بر فایل "macsvc"، یک فایل دیگر با نام "afpscan" دانلود می‌شود که تلاش می‌کند تا به دستگاه‌های موجود در شبکه متصل گردد. (شکل۸)

```
afpscan SHA256: bbbf73741078d1e74ab7281189b13f13b50308cf03d3df34bc
9f6a90065a4a55
```

**شکل (۸)**

حضور دستورات پوسته لینوکس در اسکریپت اصلی منجر به اجرای این بدافزار مخرب بر روی یک ماشین لینوکس می‌شود که با بررسی‌های انجام شده مشخص می‌گردد که به استثنای پرونده باینری Mach-O، همه اجزا بدافزار به خوبی اجرا می‌شوند. این امر نشان می‌دهد که ممکن است نوع دیگری از این بدافزار وجود داشته باشد که برای اجرا شدن بر روی لینوکس طراحی شده است که ممکن است به جای فایل باینری Mach-O از فایل دیگری استفاده کند. البته این نظر در حد فرضیه می‌باشد و فعلا مشابه این بدافزار برای سیستم‌های لینوکسی کشف نشده است.

نمونه‌های دیگری وجود دارد که نشان می‌دهد این بدافزار برای یک مدت طولانی به صورت غیر قابل تشخیص به فعالیت خود ادامه می‌دهد. در یکی از سیستم‌های مک آلوده، فایل عامل راه اندازی، یک تاریخ ایجاد داشت که مربوط به ژانویه سال ۲۰۱۵ بود. این شاهد قوی از تاریخ ایجاد درست نیست، هر چند، آن تاریخ‌ها به راحتی می‌توانند تغییر یابند. علاوه بر این یک کامنت در کد فایل macsvc وجود دارد که نشان می دهد که تغییر برای نسخه Yosemete سیستم عامل مک (Mac OS X ۱۰,۱۰)که در اکتبر ۲۰۱۴ منتشر شده ، ایجاد گردیده است .

```
if(/_(tcp|udp)\S*\s+(_\S+)$/){ $s="$2._$1"; }
elsif(/icloud\.com\.\s+(_[^\.]+\._(tcp|udp))\.\d+\.members\.btmm$/)
   { $s=$1; } # changed in yosemite
elsif(/icloud\.com\.\s+\.\s+_autotunnel6$/){ next; }
```

**شکل (۹)**

# ۴ راهکار

با توجه به قدیمی بودن کدهای استفاده شده در این بدافزار محققان امنیتی معتقدند که با یک نگاه به ماشین آلوده به راحتی می‌توان بدافزار را تشخیص داد و از روی سامانه حذف کرد و از آنتی ویروس‌های معمول و شناخته شده مانند Eset، mcafee برای کشف و حذف این بدافزار استفاده کرد.

# ۵ منابع

[۱] https://www.exploit-db.com/exploits/۳۷۲۳۸/

[۲] http://magnetco.ir
[۳] https://www.checkpoint.com/
[۴] https://www.wikipedia.org/
[۵] https://www.malwarebytes.com