Operation Ephemeral Hydra: IE Zero-Day Linked to DeputyDog Uses Diskless Method

Recently, we discovered a new IE zero-day exploit in the wild, which has been used in a strategic Web compromise. Specifically, the attackers inserted this zero-day exploit into a strategically important website, known to draw visitors that are likely interested in national and international security policy. **We have identified relationships between the infrastructure used in this attack and that used in Operation DeputyDog**. Furthermore, the attackers **loaded the payload used in this attack directly into memory without first writing to disk** – a technique not typically used by advanced persistent threat (APT) actors. This technique **will further complicate network defenders' ability to triage compromised systems, using traditional forensics methods.**

Enter Trojan.APT.9002

On November 8, 2013 our colleagues **Xiaobo Chen** and **Dan Caselden** posted about a **new Internet Explorer 0-day exploit** seen in the wild. This exploit was seen used in a strategic Web compromise. The exploit chain was limited to one website. There were no iframes or redirects to external sites to pull down the shellcode payload.

Through the FireEye Dynamic Threat Intelligence (DTI) cloud, we were able to retrieve the payload dropped in the attack. This payload has been identified as a variant of Trojan.APT.9002 (aka Hydraq/McRAT variant) and runs in memory only. **It does not write itself to disk, leaving little to no artifacts that can be used to identify infected endpoints.**

Specifically, the payload is shellcode, which is decoded **and directly injected into memory after successful exploitation** via a series of steps. After an initial XOR decoding of the payload with the key "0x9F", an instance of rundll32.exe is launched and injected with the payload using CreateProcessA, OpenProcess, VirtualAlloc, WriteProcessMemory, and CreateRemoteThread.

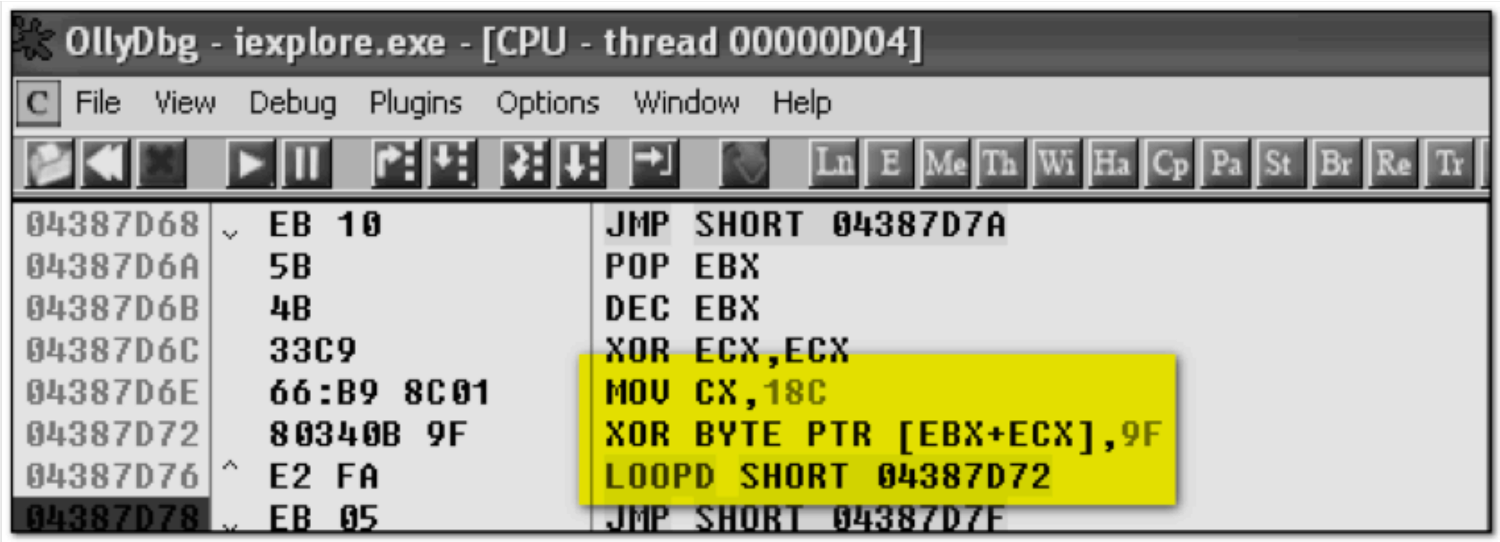
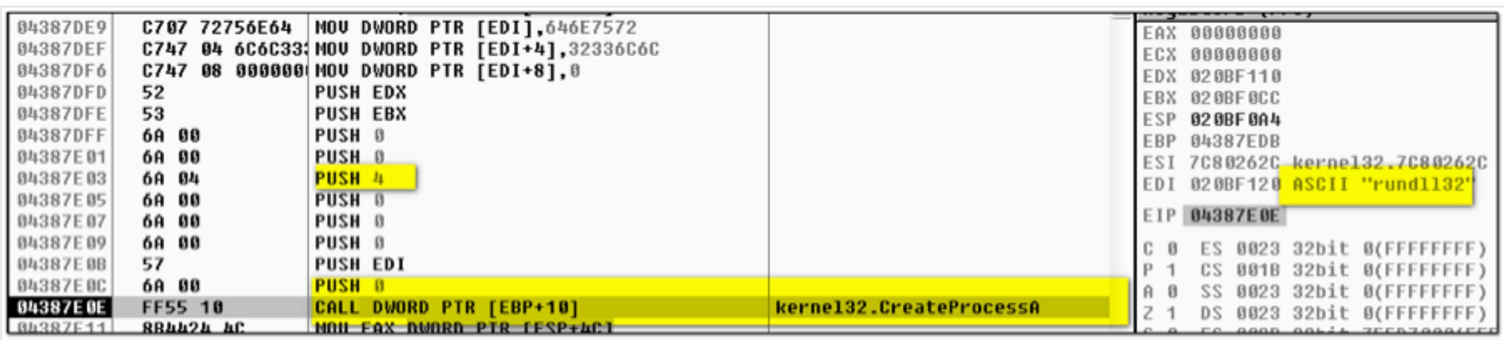
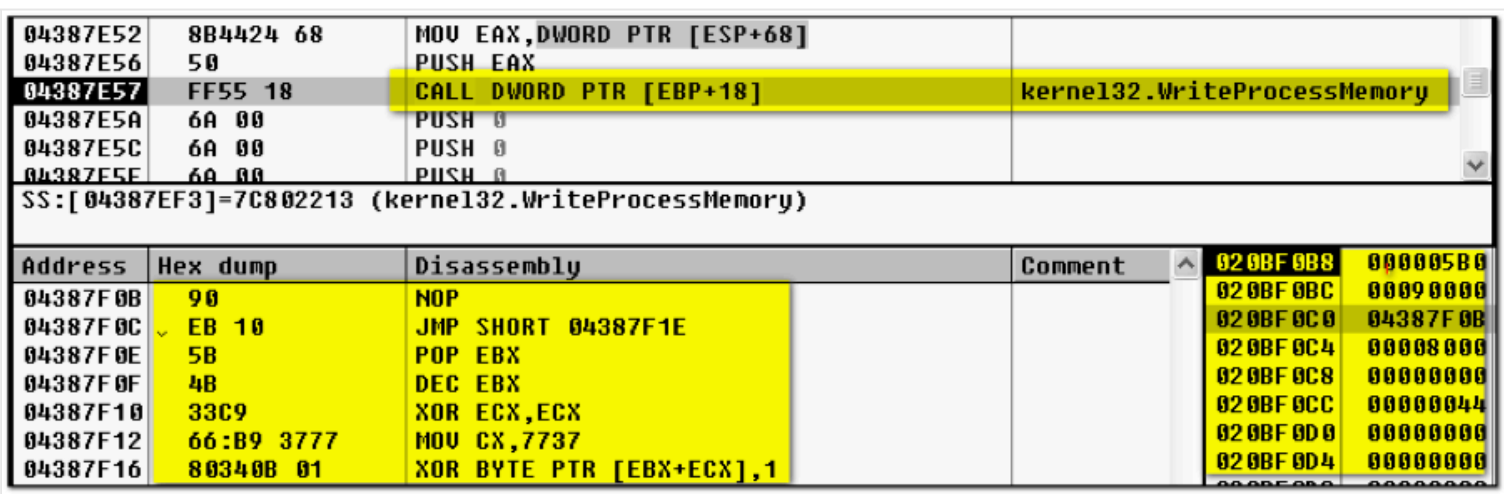


Figure 1 – Initial XOR decoding of shellcode, with key ’0x9F’





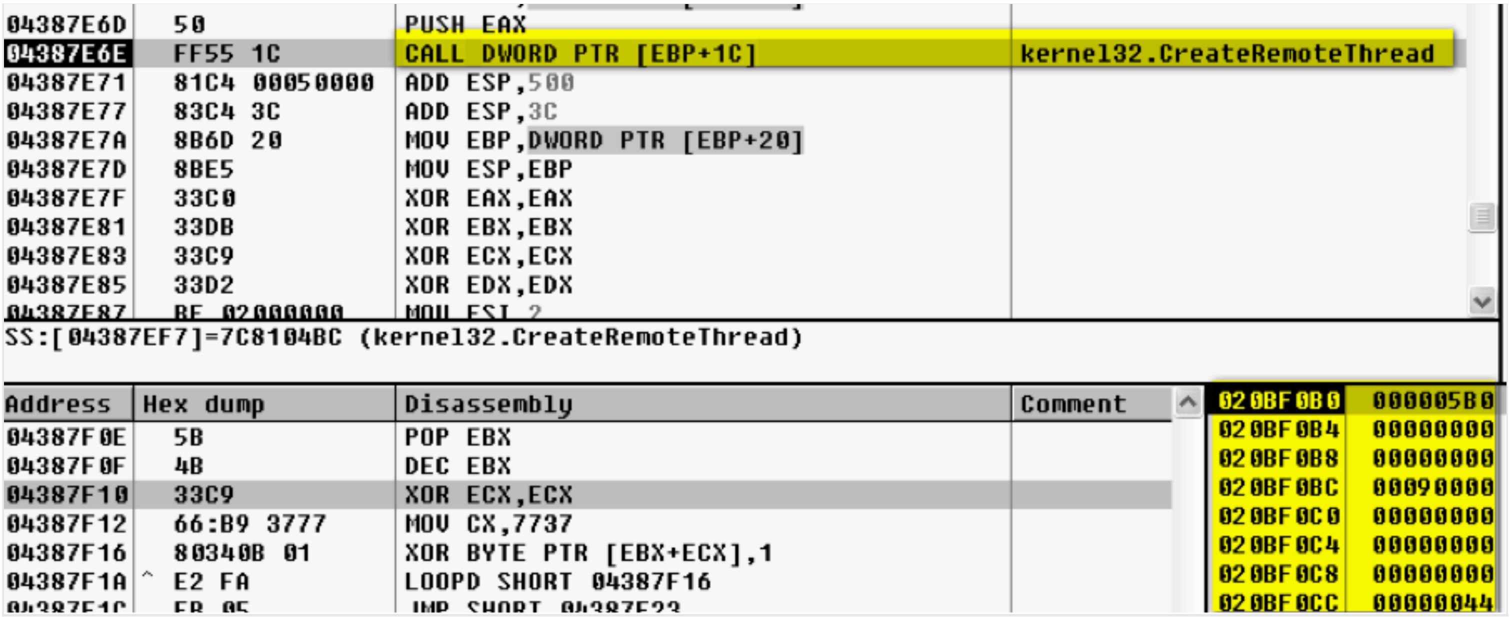


Figure 2 – Shellcode launches rundll32.exe and injects payload

After transfer of control to the injected payload in rundll32.exe, the shellcode is then subjected to two more levels of XOR decoding with the keys ’0×01′, followed by ’0x6A’.

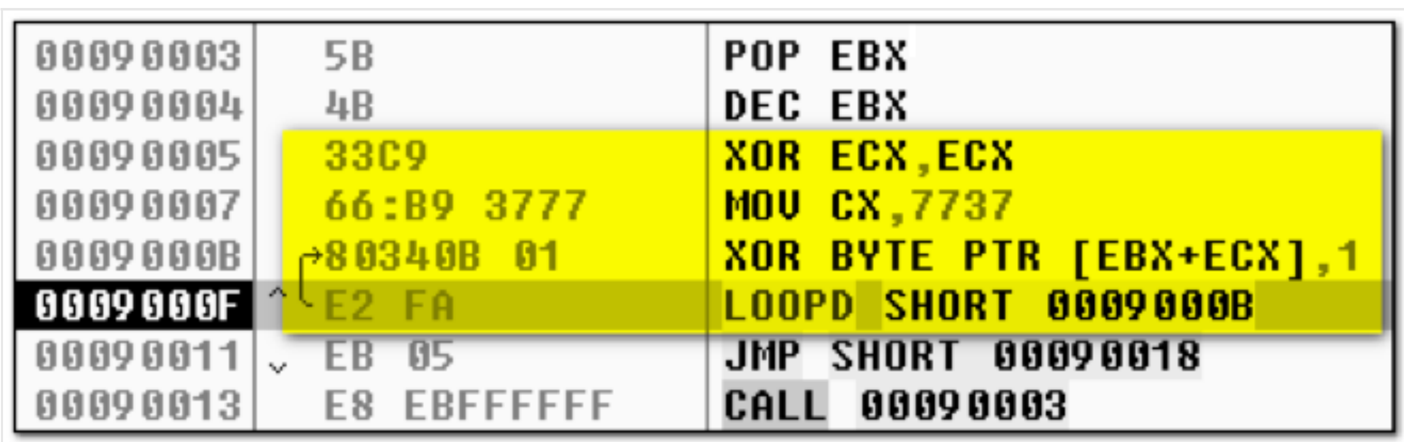


Figure 3- Decoding shellcode with XOR key ’0×01′

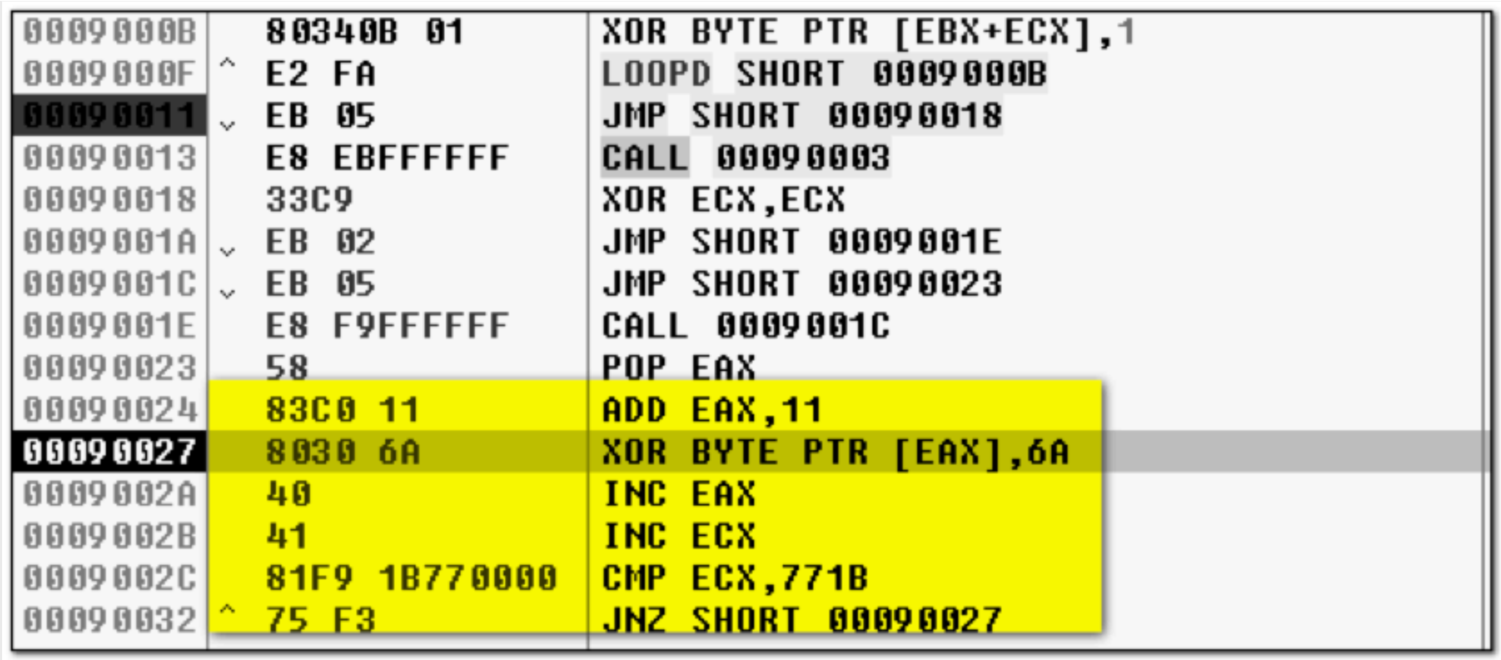


Figure 4 – Decoding shellcode with XOR key '0x6A'

Process execution is then transferred to the final decoded payload, which is a variant of the 9002 RAT.

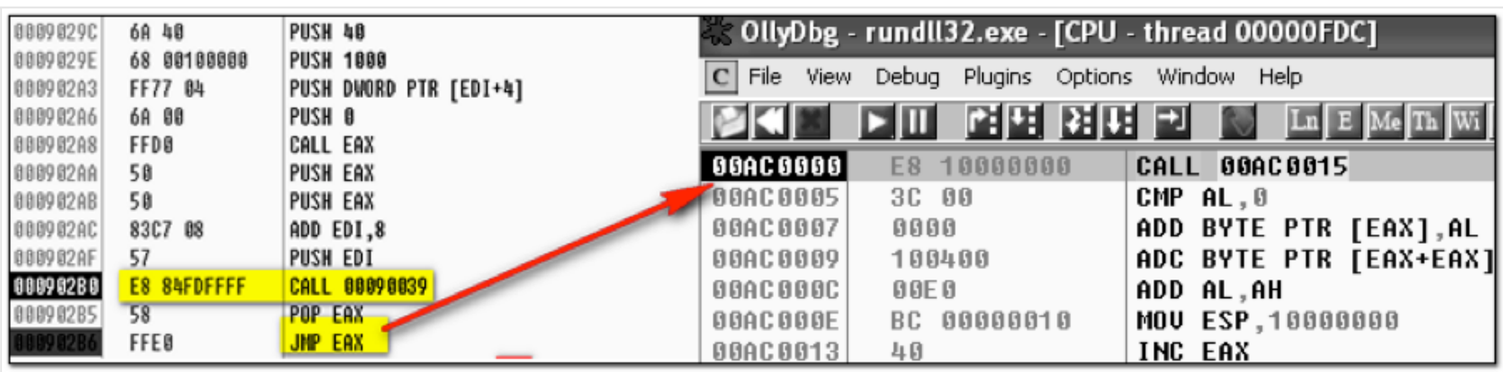


Figure 5 – Transfer of process execution to final decoded payload

The fact that the attackers used a non-persistent first stage payload suggests that they are confident in both their resources and skills. As the payload was not persistent, **the attackers had to work quickly**, in order to gain control of victims and move laterally within affected organizations. **If the attacker did not immediately seize control of infected endpoints, they risked losing these compromised endpoints**, as the endpoints could have been **rebooted** at any time – **thus automatically wiping the in-memory Trojan.APT.9002 malware variant from the infected endpoint.**

Alternatively, the use of this non-persistent first stage may suggest that the attackers were confident that their intended targets would simply revisit the compromised website and be re-infected.

Command and Control Protocol and Infrastructure

This Trojan.APT.9002 variant connected to a command and control server at 111.68.9.93 over port 443. **It uses a non-HTTP protocol as well as an HTTP POST for communicating with the remote server.** However, the callback beacons have changed in this version, in comparison to the older 9002 RATs.

The older traditional version of 9002 RAT had a static 4-byte identifier at offset 0 in the callback network traffic. This identifier was typically the string "9002", but we have also seen variants, where this has been modified – such as the 9002 variant documented in the **Sunshop campaign**.

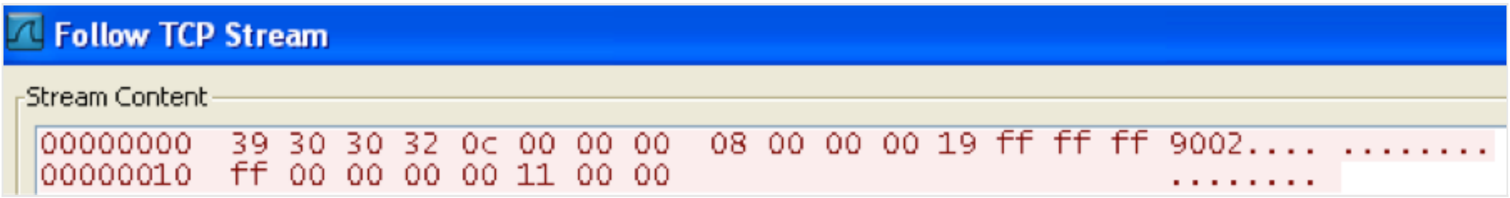


Figure 6 – Traditional 9002 RAT callback beacon

In contrast, the beacon from the **diskless 9002 payload** used in the current IE 0-day attack is **remarkably different** and uses a dynamic 4-byte XOR key to encrypt the data. This 4-byte key is present at offset 0 **and changes with each subsequent beacon. FireEye labs is aware that the 4-byte XOR version of 9002 has been in the wild for a while and is used by multiple APT actors, but this is the first time we've seen it deployed in the diskless payload method.**

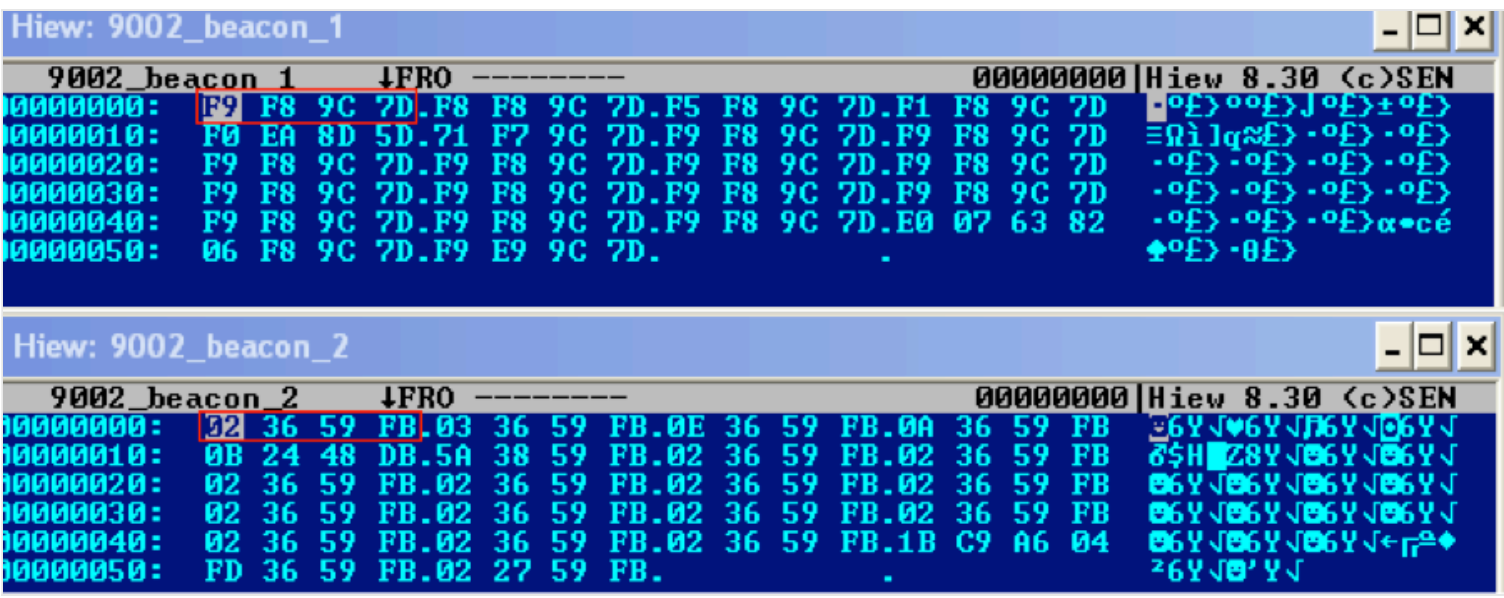


Figure 7 – Sample callback beacons of the diskless 9002 RAT payload

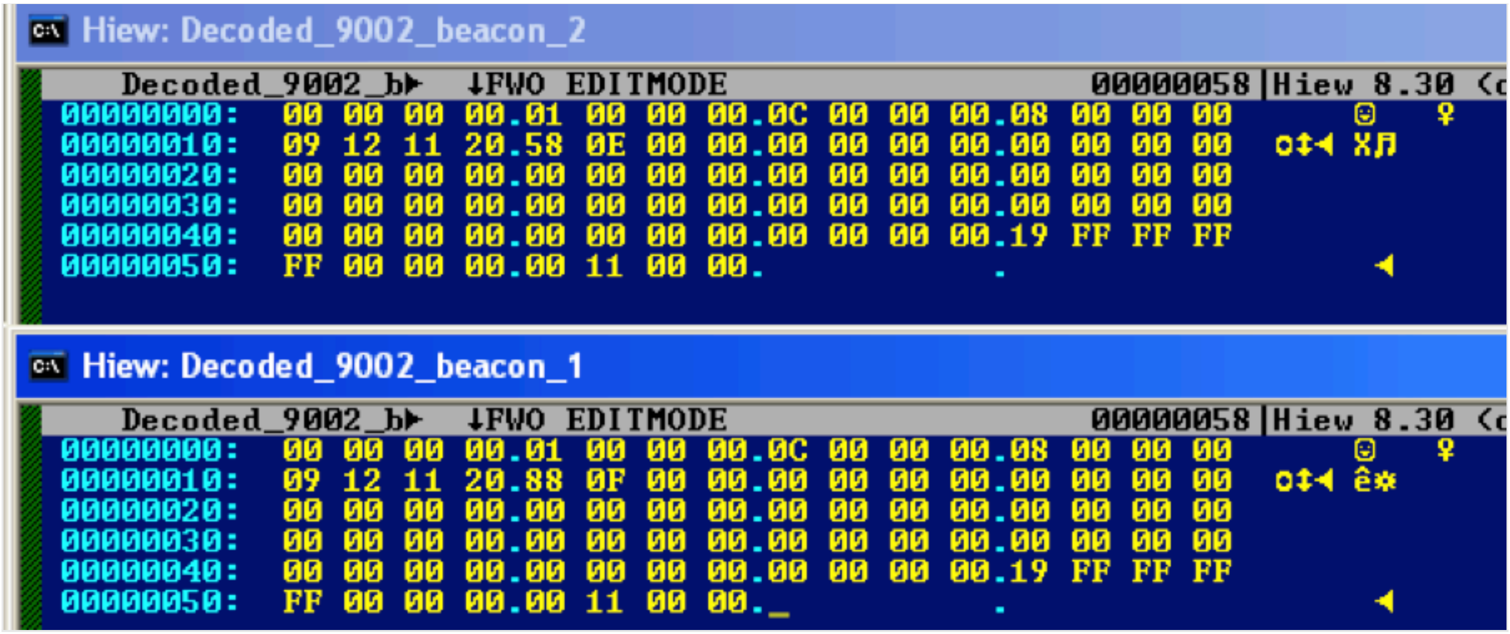


Figure 8 – XOR decrypted callback beacons of the diskless 9002 RAT payload

The XOR decoded data always contains the static value "\x09\x12\x11\x20" at offset 16. This value is in fact hardcoded in packet data construction function prior to XOR encoding. This value most likely is the date "2011-12-09" but its significance is not known at this time.

```
push    eax
push    0
lea     ecx, [ebp+var_54]
push    ecx
call    sub_E16C
add     esp, 0Ch
lea     edx, [ebp+var_470]
push    edx
call    dword ptr ds:402D58h
mov     eax, [ebp+var_470]
mov     [ebp+var_54], eax
mov     [ebp+var_44], 20111209h
mov     [ebp+var_50], 0
mov     [ebp+var_4C], 0
mov     [ebp+var_48], 0
mov     ecx, ds:41085Ch
mov     [ebp+var_40], ecx
mov     edx, [ebp+var_54]
push    edx
mov     eax, [ebp+var_460]
sub     eax, 4
push    eax
lea     ecx, [ebp+var_50]
push    ecx
call    sub_D2B1        ; XOR_ENC_FUNC
```

Figure 9 – Packet data construction function showing hardcoded value

The diskless 9002 RAT payload also makes a POST request, which has also changed from the traditional version. It has Base64 stub data, instead of the static string “AA”. **The User-Agent string and URI pattern remain the same however.** It uses the static string “lynx” in the User-Agent string and the URI is incremental hexadecimal values.

Traditional 9002 RAT	**Diskless 9002 RAT**
POST /4 HTTP/1.1 **User-Agent: lynx** Host: ieee.boeing-job.com Content-Length: 2 Connection: Keep-Alive Cache-Control: no-cache	POST /2 HTTP/1.1 **User-Agent: lynx** Host: 111.68.9.93:443 Content-Length: 104 Connection: Keep-Alive Cache-Control: no-cache

AA	wUeAKsFHgCrBR4AqwUeAKshVkQrBR4Aqw UeAKsFHgCrBR4AqwUeAKsFHgCrBR4Aqw UeAKsFHgCrBR4AqwUeAKsFHgCrBR4AqwUe AKg==

The data in the POST stub is also encrypted with a 4-byte XOR key, and when decrypted, the data is similar to the data in the non-HTTP beacon and also has the static value "\x09\x12\x11\x20".

Campaign Analysis

We previously observed 104130d666ab3f640255140007f0b12d connecting to the same 111.68.9.93 IP address.

Analysis of MD5 104130d666ab3f640255140007f0b12d revealed that it shared unique identifying characteristics with 90a37e54c53ffb78969644b1a7038e8c, acbc249061a6a2fb09271a68d53567d9, and 20854f54b0d03118681410245be39bd8.

MD5 acbc249061a6a2fb09271a68d53567d9 and 90a37e54c53ffb78969644b1a7038e8c are both Trojan.APT.9002 variants and connect to a command and control server at 58.64.143.244.

MD5 20854f54b0d03118681410245be39bd8 is another Trojan.APT.9002 variant. This variant connected to a command and control server at ad04.bounceme.net.

Passive DNS analysis of this domain revealed that it resolved to 58.64.213.104 between 2011-09-23 and 2011-10-21. The following other domains have also been seen resolving to this same IP address:

DOMAIN	FIRST SEEN	LAST SEEN
dll.freshdns.org	2011-12-08	2012-01-31
grado.selfip.com	2011-12-23	2012-01-10
usc-data.suroot.com	2012-02-20	2012-02-22
usa-mail.scieron.com	2011-12-01	2012-02-22

If the domain dll.freshdns.org rings a bell, it should. While covering a different Internet Explorer Zero-day (CVE-2013-3893) and the associated Operation DeputyDog campaign, we reported that the CnC infrastructure used in that campaign overlapped with this same domain: dll.freshdns.org.

Inside the in-memory version of the Trojan.APT.9002 payload used in this strategic Web compromise, we identified the following interesting string: "rat_UnInstall". **Through DTI, we found this same string present in a number of different samples including the ones discussed above:**

104130d666ab3f640255140007f0b12d
90a37e54c53ffb78969644b1a7038e8c
acbc249061a6a2fb09271a68d53567d9

20854f54b0d03118681410245be39bd8

Based on this analysis, all of these samples, including the in-memory variant, can be detected with the following simple YARA signature:

```
rule FE_APT_9002_rat
{
  meta:
    author = "FireEye Labs"
  strings:
    $mz = {4d 5a}
    $a = "rat_UnInstall" wide ascii
  condition:
    ($mz at 0) and $a
}
```

We also found the following strings of interest present in these above 9002 RAT samples (excluding the in-memory variant):

McpRoXy.exe
SoundMax.dll

These strings were all observed and highlighted by Bit9 **here**. **As Bit9 notes in their blog, Trojan.APT.9002 (aka Hydraq/McRAT) was also used in the original Operation Aurora campaign, and the "rat_UnInstall" string can be found in the original Aurora samples confirming the lineage.**

Conclusions

By utilizing strategic Web compromises along with in-memory payload delivery tactics and multiple nested methods of obfuscation, this campaign has proven to be exceptionally accomplished and elusive. APT actors are clearly learning and employing new tactics. **With uncanny timing and a penchant for consistently employing Zero-day exploits in targeted attacks, we expect APT threat actors to continue to evolve and launch new campaigns for the foreseeable future.** Not surprisingly, these old dogs continue to learn new tricks.

FireEye Labs would like to thank iSIGHT Partners for their assistance with this research.

This entry was posted in Exploits, Targeted Attack, Threat Intelligence, Threat Research, Vulnerabilities by Ned Moran, Sai Omkar Vashisht, Mike Scott and Thoufique Haq. Bookmark the permalink.